

95



**UNIVERSIDADE FEDERAL DE SANTA CATARINA
COORDENADORIA DO CURSO DE PÓS-GRADUAÇÃO
EM ENGENHARIA DE PRODUÇÃO**

TESE DE DOUTORADO

**UM SISTEMA INTELIGENTE DE SIMULAÇÃO PARA A AVALIAÇÃO
DE DESEMPENHO
OPERACIONAL DE SISTEMAS FLEXÍVEIS DE MANUFATURA**

**DOUTORANDO
PAULO JOSÉ DE FREITAS FILHO**

**ORIENTADOR
CRISTIANO A. J. CUNHA, Dr**

UM SISTEMA INTELIGENTE DE SIMULAÇÃO PARA A AVALIAÇÃO OPERACIONAL DE SISTEMAS FLEXÍVEIS DE MANUFATURA

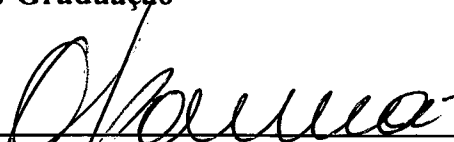
PAULO JOSÉ DE FREITAS FILHO

Esta tese foi julgada adequada para a obtenção do título de

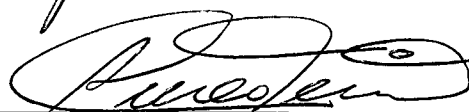
“Doutor em Engenharia”

Especialidade em Engenharia de Produção e aprovada em sua forma final pelo
Programa de Pós-Graduação

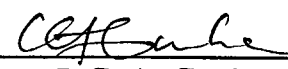
Banca Examinadora:



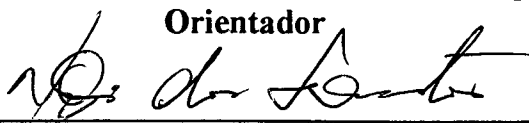
Prof. Osmar Possamai, Dr. Ing.
Coordenador do Curso



Prof. Áureo C. Ferreira, Ph. D.
Moderador




Prof. Cristiano J. C. A. Cunha, Dr. rer. pol
Orientador



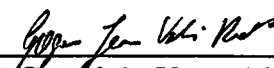
Prof. Neri dos Santos, Dr. Ing.



Prof. Ricardo Miranda Barcia, Ph. D.



Prof. Edgard Dias Batista Jr., Dr.



Prof. Gregório Varvakis Rados, Ph. D.

**À minha mulher Branca
e às minhas filhas Mariana e Manuela
por todo o amor, apoio e incentivo**

AGRADECIMENTOS

Manifesto meus sinceros agradecimentos às seguintes pessoas e instituições:

- Ao Prof. Cristiano Cunha, Dr., pela brilhante e dedicada orientação ao longo de todo este trabalho;
- Ao Prof. Suresh K. Khator, Ph.D., que durante meu período de estágio e aperfeiçoamento na University of South Florida, Tampa, USA, muito contribuiu para o desenvolvimento deste trabalho, com seus conhecimentos e preciosos conselhos;
- Aos professores doutores integrantes da Banca Examinadora, pelos valiosos comentários e sugestões, sem os quais este trabalho não seria aperfeiçoado;
- À CAPES, pelo auxílio financeiro;
- Aos professores e funcionários do Departamento de Informática e Estatística da UFSC, por todo o apoio e incentivo demonstrado;
- A todas as pessoas que de forma direta ou indireta, contribuíram para a realização deste trabalho.

SUMÁRIO

LISTA DE FIGURAS	ix
LISTA DE GRÁFICOS	xii
LISTA DE TABELAS	xiii
RESUMO	xiv
ABSTRACT	xv
 CAPÍTULO I - INTRODUÇÃO	 1
1.1. Origem do Trabalho	3
1.2. Definição do Problema	4
1.3. Objetivos do Trabalho	6
1.4. Importância do Trabalho	8
1.5. Suposições, Alcance e Limitações do Trabalho	9
1.6. Organização do Trabalho	10
 CAPÍTULO II - REVISÃO BIBLIOGRÁFICA	 12
2.1. Flexibilidade e Sistemas de Produção	12
2.1.1. O Conceito e Importância da Flexibilidade na Manufatura	13
2.1.2. Tipos de Flexibilidades	14
2.2. Sistemas Flexíveis de Manufatura	15
2.2.1 Componentes dos Sistemas Flexíveis de Manufatura	17
2.2.1.1 Centros de Usinagem Numericamente Controlados	18
2.2.1.2 Sistemas Automáticos de Transporte e Manuseio de Materiais	18
2.2.1.3 Controle Computacional	19
2.3 Planejamento e Controle Operacional de SFM	21
2.3.1 Planejamento da Operação	21
2.3.2 Controle da Operação	23
2.3.2.1 Despacho de Peças	23
2.3.2.2 Despacho de Veículos	25

2.3.2.3	Impacto dos Parâmetros do Sistema	27
2.4.	Avaliação de Projetos de Sistemas Flexíveis de Manufatura	28
2.4.1	Medidas de Desempenho	29
2.4.2	Modelos para a Avaliação de SFM	29
2.5.	Simulação de Sistemas Flexíveis de Manufatura	32
2.5.1	Simulação Aplicada ao Desenvolvimento de Projetos de SFM	34
2.5.2	Simulação Aplicada ao Planejamento e Controle da Produção de SFM	35
2.5.3.	Avanços no Uso da Simulação na Modelagem de SFM	37
2.6.	Sistemas Especialistas	38
2.6.1	Definição	38
2.6.2	Diferenças entre Sistemas Especialistas e a Programação Convencional	39
2.6.3.	Estrutura de um Sistema Especialista	40
2.6.3.1	A Base de Conhecimentos	40
2.6.3.2	A Máquina de Inferência	42
2.6.4.	Ferramentas para o Desenvolvimento de Sistemas Especialistas	46
2.7	Sistemas Inteligentes de Simulação (SIS)	47
2.7.1	As Similaridades entre Sistemas Especialistas e Simulação	48
2.7.2.	O Uso de Sistemas Inteligentes de Simulação em SFM	51
2.8	Sumário	54

CAPÍTULO III - MODELO DE SIMULAÇÃO COM ALOCAÇÃO DINÂMICA DE MÁQUINAS PARA O PLANEJAMENTO E CONTROLE OPERACIONAL DE SFM

3.1	Planejamento da Produção Vs Controle Operacional em SFM	55
3.1.1	Elementos do Planejamento da Produção em SFM	56
3.1.1.1	Hipóteses Associadas aos Modelos de Planejamento de SFM	57
3.1.2	Elementos do Controle Operacional de SFM	58
3.1.2.1	Regras para a Liberação de Novas Peças ao Sistema	59
3.1.2.2	Regras para a Priorização de Peças no Buffer das Máquinas	60
3.1.2.3	Políticas para o Despacho de Veículos Transportadores	61

3.1.3	Parâmetros Secundários do Sistema	62
3.2	Modelo para o Planejamento e Controle Operacional de SFM	62
3.2.1	Parâmetros Utilizados no Modelo	63
3.2.2	Modelo de Simulação Proposto	65
3.2.2.1	Controle Lógico da Seqüência de Visitações de Máquinas	67
3.2.2.2	Hipóteses Associadas ao Modelo	70
3.2.2.3	Controle Lógico do Fluxo de Peças	71
3.2.2.4	Controle Lógico das Máquinas	72
3.2.2.5	Controle Lógico do Fluxo dos VAGs	73
3.3.	Medidas de Desempenho	75
3.4	Sumário	76

CAPÍTULO VI - APLICAÇÃO DO MODELO DE SIMULAÇÃO PRO - POSTO E COMPARAÇÃO DOS RESULTADOS COM OS DOS MODELOS TRADICIONAIS

4.1	Características do Sistema Flexível de Manufatura Modelado	78
4.2	Validação do Sistema Flexível de Manufatura Modelado	79
4.2.1	Problema Utilizado na Validação	80
4.2.2	Os Resultados Obtidos	81
4.2.3	Testes de Hipóteses para as Médias Obtidas	82
4.3	Aplicação do Modelo Proposto a Problemas de Planejamento e Controle Operacional de SFM	83
4.3.1	Geração Numérica de Problemas Exemplos	84
4.3.2	Comparação de Resultados	89
4.3.2.1	Experimentos Realizados	91
4.3.2.2	Problema Exemplo Utilizado na Comparação de Resultados	92
4.3.2.3	Medidas de Desempenho Adotadas na Comparação dos Resultados	92
4.3.2.4	Resultados Obtidos nas Comparações	93
4.3.2.5	Observações sobre os Resultados Obtidos	94
4.3.2.6	Testes de Hipóteses para Comparação das Médias Obtidas nos Experimentos	95
4.3.3	Análise de Sensibilidade do Modelo	97

4.3.3.1	Observações sobre os Resultados das Simulações	98
4.3.4	Busca das Melhores Combinações entre os Elementos Operacionais	102
4.3.4.1	Resultados e Comentários sobre os Novos Experimentos Realizados	102
4.4	Sumário	112

CAPÍTULO V - ESTRUTURA DO PROTÓTIPO DO SISTEMA INTELIGENTE DE SIMULAÇÃO

113

5.1	Objetivo do Protótipo	113
5.2	Estrutura do Protótipo	114
5.2.1	Interface Gráfica Usuário-Sistema (GUI)	115
5.2.2	Módulo de Simulação	116
5.2.3	Módulo de Análise	119
5.3	O Sistema de Decisão	121
5.3.1	Comunicação entre o Módulo de Simulação e o Módulo de Análise	122
5.3.2	A Estrutura de Decisão	
5.3.2.1	Classes, Objetos, Frames, Slots e Métodos	124
5.3.2.2	Regras de Produção	129
5.3.3	Processo de Análise Executado pelo Módulo de Análise	130
5.4	Sumário	133

CAPÍTULO VI - APLICAÇÃO E DEMONSTRAÇÃO DE USO DO PROTÓTIPO

134

6.1	O Procedimento Interativo	135
6.2	Aplicação do Protótipo e seus Resultados	136
6.2.1	Solução Inicial	136
6.2.2	Simulação e Análise dos Resultados da Solução Inicial	137
6.2.3	Simulação e Análise dos Resultados das Interações Subsequentes	129
6.3	Comentários sobre os Resultados, Procedimento Interativo e Uso do Protótipo	148
6.3.1	Comentários sobre os Resultados	148
6.3.2	Comentários sobre o Procedimento Interativo	150

6.3.3	Comentários sobre o Uso do Protótipo	150
6.4	Sumário	151
CAPÍTULO VII - CONCLUSÕES E RECOMENDAÇÕES À FUTURAS PESQUISAS		152
7.1	Resultados e Contribuições da Pesquisa	152
7.2	Limitações do Trabalho	156
7.3	Recomendações à Futuras Pesquisas	157
7.3.1	Extensão da Capacidade do Módulo de Análise	157
7.3.2	Extensão da Capacidade do Módulo de Simulação	158
7.3.3	Integração do Fluxo de Ferramentas ao Modelo.	158
7.3.4	Outras Extensões	159
ANEXOS		160
ANEXO 1	Código Fonte do Programa para Alocação Dinâmica dos Recursos em Tempo de Simulação	161
ANEXO 2	Código Fonte de todas as Regras do Protótipo	168
ANEXO 3	Principais Telas do Protótipo	181
ANEXO 4	Código Fonte do Modelo de Simulação em SIMAN	195
ANEXO 5	Código Fonte do Experiment File em SIMAN IV	199
ANEXO 6	Código Fonte do Protótipo com todas as Classes, Objetos, Instâncias, Funções e Métodos	201
REFERÊNCIAS BIBLIOGRÁFICAS		233

LISTA DE FIGURAS

Figura 2.1	<u>Um Exemplo de Sistema Flexível de Manufatura</u>	17
Figura 3.1	Elementos dos Modelos de Planejamento de SFM	56
Figura 3.2	Elementos do Controle Operacional de SFM	59
Figura 3.3	Controle Lógico da Seqüência de Visitações de Máquinas	68
Figura 3.4	Controle do Fluxo de Peças	72
Figura 3.5	Controle Lógico das Máquinas	73
Figura 3.6	Controle Lógico do Fluxo de VAGs	74
Figura 4.1	Layout do SFM Exemplo baseado no modelo citado por CARRIE	78
Figura 4.2	Visão Esquemática do Processo de Geração de Problemas Exemplos	86
Figura 4.3	Representação Esquemática das duas Abordagens	91
Figura 5.1	Estrutura Geral do Protótipo	114
Figura 5.2	Tela Inicial do Protótipo com Alguns dos Elementos da GUI	115
Figura 5.3	Tela Inicial do Módulo de Simulação com suas Duas opções	117
Figura 5.4	Estrutura do Módulo de Simulação	118
Figura 5.5	Tela Inicial do Módulo de Análise	119
Figura 5.6	Janela com os resultados da Simulação Referentes as Peças Produzidas	120
Figura 5.7	Estrutura do Módulo de Análise dos Resultados da Simulação	121
Figura 5.8	Processo de comunicação entre módulos de simulação e análise por meio de arquivos	122
Figura 5.9	Elementos Básicos do MS e do MA	123
Figura 5.10	Algumas Classes e Instâncias do Protótipo	125
Figura 5.11	A Classe de Máquinas Exibindo a Lista Parcial de seus Atributos e a Lista de Métodos Associados a Alguns Deles.	126
Figura 5.12	O Método Diag_Util_Maq associado ao Slot UtilMedMaq	128
Figura 5.13	Inter-relações entre Alguns Métodos e Conjuntos de Regras para a Realização do Diagnóstico Operacional das Máquinas	131
Figura 5.14	Exemplo de uma Regra de Otimização do Sistema	132
Figura 6.1	Estágios de tomada de decisão no Procedimento Iterativo	135
Figura 6.2	Resultados das análises realizadas pelo MA para a simulação da solução inicial	140

Figura 6.3	Tela com os resultados das análises da interação 3 elaboradas pelo MA	143
Figura 6.4	Tela com os resultados das análises da interação 4 elaboradas pelo MA	144
Figura 6.5	Tela com os resultados das análises da interação 5 elaboradas pelo MA	146
Figura 6.6	Tela com os resultados das análises da interação 6 elaboradas pelo MA	148
Figura A.1	Tela Inicial do Protótipo com alguns Elementos da Interface Gráfica	181
Figura A.2	Tela Inicial do Módulo de Simulação com suas duas Opções	182
Figura A.3	Tela que Permite ao Usuário a Definição dos Parâmetros e Políticas Operacionais a Serem Empregados durante a Simulação do Sistema	183
Figura A.4	Exemplo da definição de um cenário. Nesta Tela são Definidas quais peças serão fabricadas e em qual volume	184
Figura A.5	A tela acima permite a definição dos parâmetros das máquinas. Podemos definir que operações cada uma realizará, com que eficiência, as taxas de utilização máxima e mínima e o tamanho dos buffers.	185
Figura A.6	Na tela acima apresentamos um exemplo de definição de políticas operacionais. No caso, a política de despacho de peças	186
Figura A.7	A tela acima mostra um exemplo de apresentação dos resultados da simulação. No caso os resultados reference as peças produzidas	187
Figura A.8	Tela de apresentação dos resultados simulação referente ao desempenho das máquinas	188
Figura A.9	Tela de apresentação dos resultados simulação referente ao desempenho dos VAGs	189
Figura A.10	Tela de apresentação dos resultados simulação mostrando duas das principais medidas de desempenho: Tempo Médio do Ciclo de Produção e Tempo Médio do Fluxo de Peças	190
Figura A.11	Tela inicial do Módulo de Análise. A última opção permite iniciar o processo de análise utilizando o MA	191
Figura A.12	A tela acima permite ao usuário definir alguns parâmetros básicos para iniciar o processo de análise	192
Figura A.13	A tela acima é um exemplo dos resultados das análises efetuadas pelo MA. Neste exemplo, o MA sugere a continuidade das simulações.	193

com alterações nos parâmetros do modelo de simulação.

Figura A.14 A tela acima é outro exemplo dos resultados das análises efetuadas 194 pelo MA. Nesta, o módulo de análise está sugerindo o encerramento do processo iterativo.


LISTA DE GRÁFICOS

Gráfico 4.1	Ganhos percentuais obtidos pelo uso do novo modelo	95
Gráfico 4.2	Efeito das regras para a liberação de peças	98
Gráfico 4.3	Efeito das regras para o despacho de veículos	98
Gráfico 4.4	Efeito do tamanho dos Buffers	99
Gráfico 4.5	Efeito do número de Pallets	99
Gráfico 4.6	Resultados para o uso de 2 VAGs	103
Gráfico 4.7	Resultados para o uso de 3 VAGs	104
Gráfico 4.8	Resultados para o uso de 4 VAGs	104
Gráfico 4.9	Resultados para o uso de 2 VAGs	105
Gráfico 4.10	Resultados para o uso de 3 VAGs	106
Gráfico 4.11	Resultados para o uso de 4 VAGs	106
Gráfico 4.12	Resultados para o uso de 2 VAGs	108
Gráfico 4.13	Resultados para o uso de 3 VAGs	108
Gráfico 4.14	Resultados para o uso de 4 VAGs	109
Gráfico 4.15	Resultados para o uso de 2 VAGs	110
Gráfico 4.16	Resultados para o uso de 3 VAGs	111
Gráfico 4.17	Resultados para o uso de 4 VAGs	111

LISTA DE TABELAS

Tabela 2.1	Hierarquia e funções de controle computacional em um SFM	20
Tabela 2.2	Medidas de Desempenho	29
Tabela 4.1	Mix de Produção Desejado	80
Tabela 4.2	Seqüências Operacionais e Respectivos Tempos	80
Tabela 4.3	Resultados dos Dez Experimentos com o Modelo em SIMAN	81
Tabela 4.4	Resultados do Experimento com o Modelo em ECSL	82
Tabela 4.5	Resultados dos Testes de Hipóteses para as Médias	83
Tabela 4.6	Operações e Tempos Necessários aos Vários Tipos de Peças Produzidas	87
Tabela 4.7	Eficiências na Realização das Operações pelas Máquinas do SFM	88
Tabela 4.8	Um Conjunto de Problemas Criados pela Execução da Segunda Etapa do Processo de Geração de Problemas Exemplos	89
Tabela 4.9	Resultados dos Experimentos quando se Aplicam as Duas Abordagens	93
Tabela 4.10	Resultados dos Testes de Comparação das Médias de TCP e Tfluxo	96
Tabela 6.1	Solução inicial (parâmetros e regras) para o problema exemplo	137
Tabela 6.2	Resultados da Solução Inicial Referentes as Peças	138
Tabela 6.3	Resultados da Solução Inicial Referentes as Máquinas	138
Tabela 6.4	Resultados da Solução Inicial Referentes as VAGs	138
Tabela 6.5	Resultados da Solução Inicial Referentes ao Sistema	138
Tabela 6.6	Resultados da segunda Simulação	141
Tabela 6.7	Resultados da terceira Simulação	142
Tabela 6.8	Resultados da quarta Simulação	143
Tabela 6.9	Resultados da quinta Simulação	145
Tabela 6.10	Resultados da sexta Simulação	146
		133

RESUMO



A adaptação às novas tecnologias de fabricação e às novas condições de mercado, caracterizadas pela diminuição do tempo de inovação, aumento da variedade dos produtos, redução do tamanho dos lotes de produção e maior exigência em relação à qualidade, tem levado as empresas a considerar a adoção de Sistemas Flexíveis de Manufatura (SFM).

O crescente interesse por sistemas de produção com alto grau de automatização, como os SFM, a clara escassez de modelos e programas computacionais voltados ao tratamento de problemas associados as fases de planejamento e controle operacional destes sistemas e, o pequeno número de ferramentas computacionais que permitam a avaliação do desempenho operacional dos SFM, foram a origem e motivação deste trabalho de pesquisa.

Dois dos principais problemas e aspectos referentes ao planejamento e controle operacional dos SFM são tratados neste trabalho. O primeiro, refere-se à falta de uma metodologia apropriada, que permita avaliar, com facilidade, as inter-relações entre a fase de planejamento e a fase de controle operacional de SFM, diante das inúmeras alternativas e estratégias a elas associadas. As abordagens atuais dissociam as fases de planejamento e operação dos SFM, com conseqüentes descompassos e a geração de soluções pobres e ineficientes. O segundo, é a pouca disponibilidade de sistemas computacionais que incorporem modelos voltados para o encaminhamento de soluções ao problema acima citado e, ao mesmo tempo, sejam suficientemente inteligentes e amigáveis para auxiliar o decisor ou analista, nesta complexa tarefa.

Para o tratamento do primeiro problema, desenvolvemos e implementamos um modelo, baseado em um programa de simulação o qual, está incorporado de um algoritmo para a alocação dinâmica dos recursos, de acordo com o estado do sistema, no momento da decisão. Este modelo, permite a avaliação do desempenho operacional de SFM, considerando as inter-relações entre as fases de planejamento e de controle operacional destes sistemas. Para permitir a plena utilização do modelo de simulação, desenvolvemos, também, um protótipo computacional para o mesmo. Esta ferramenta, incorpora o módulo de simulação de SFM, um módulo inteligente para a interpretação e análise dos resultados das simulações e uma interface gráfica com o usuário.

Ao longo do trabalho vários experimentos foram realizados, utilizando-se do modelo e do protótipo desenvolvidos, com os seguintes propósitos:

- Investigar o desempenho operacional dos SFM, diante de diferentes políticas operacionais em conjunto com diferentes valores associados aos parâmetros secundários do sistema;
- Comparar os resultados obtidos, com a abordagem proposta, com outros trabalhos de pesquisa que tratam do mesmo tema;
- Adquirir experiência e conhecimentos com o propósito de incorporá-los ao módulo inteligente para a interpretação e análise dos resultados da simulação.

Ao final do trabalho, demonstramos a aplicabilidade do protótipo do sistema inteligente de simulação, na solução de um problema exemplo. O problema, trata da busca de um melhor desempenho operacional para um SFM, diante das várias combinações de suas políticas de gerenciamento e de seus parâmetros operacionais

ABSTRACT

In now a days, many companies have been consider the adoption of Flexible Manufacturing Systems (FMS). This is specially caused by the new market conditions characterized by the reduction of the innovation time, great product variety, reduction of batch size and demand of quality.

The great concern by high automated production systems, like the FMS, the lack of models and computational programs to deal with problems of these systems and the need of computational tools that allow the assess of the FMS performance, are the origin and the main reasons for this research.

Two important aspects and problems associated to planning and operational control in FMS are addressed in this study. First, the lack of an appropriated methodology to easy treaty and assess the interrelationship between planning and operational control decisions in FMS due to many possibilities of control strategies and planning objectives. Nearly all of the current approaches treat separately the problems of this two phases generating planning solutions that are, many times, not feasible at the operational level. The second aspect is the scarcity of computational programs that incorporate models to solve the above problem and, at the same time are user friendly and have some degree of intelligence to help the analyst in this complex job.

In this study, we develop and implement a simulation model, incorporated of a dynamic allocation resource algorithm, which consider the state of the system at the decision moment. This model allow the user to assess the operational performance of the FMS, considering the relations between planning and operational levels. The computational prototype of this model, integrate the simulation tool with a intelligent module to help the results analysis and a graphical user interface.

During the study many experiments are made with the following purposes:

- Investigate the FMS performance against the many combinations of operational polices and systems parameters;
- Compare the results of the proposal approach with other studies that treat the same theme;
- Get skill and knowledge with the purpose of incorporate it to the intelligent module of the prototype.

At the ^eand of the study we made a application of the prototype addressing a solution of a FMS planning and control problem demonstrating the suitability and usefulness of the model.

CAPÍTULO I

INTRODUÇÃO

O desenvolvimento da produção industrial pode ser descrito através de três importantes câmbios estruturais. A primeira evolução industrial (Idade da Energia) iniciou no fim do século 18 com o surgimento da máquina a vapor que permitiu a substituição da força muscular. A segunda grande evolução industrial iniciou no fim do século 19. A introdução da eletricidade, juntamente com as diferentes tecnologias que permitiram a sua utilização, conduziram à descentralização das fontes propulsoras. Com isto, foi criada a base para a mecanização dos processos industriais. Este foi o início da "Idade da Mecanização". Hoje, está em andamento a "Idade da Informação" que teve seu início nos anos 50 com a introdução do computador na empresa. Esta terceira fase do desenvolvimento industrial é caracterizada pelo dramático avanço das tecnologias relacionadas com a informação. Elas determinarão o futuro da fábrica e serão a base para a administração da fábrica do futuro.

As novas tendências da informática abrem grandes possibilidades para as empresas (e os países) em condições de aproveitar as oportunidades oferecidas por estas novas tecnologias. Para outras empresas, entretanto, o ritmo acelerado da mudança tecnológica pode constituir uma ameaça. As atuais tendências mostram uma polarização, cada vez maior, entre as empresas capazes de ajustarem-se, com êxito, à nova realidade da informatização da produção e as que não podem fazê-lo.

Além disso, as novas condições do mercado - caracterizadas pela diminuição do tempo de inovação, aumento da variedade dos produtos, diminuição dos tamanhos dos lotes de produção e maior exigência em relação à qualidade - forçam uma nova tomada de posição em relação às áreas funcionais. A área de produção, que era considerada apenas como um objeto do planejamento operacional, passou a ser considerada como um importante elemento para a elaboração de estratégias empresariais.

Com isto, a idéia básica da gerência de produção, ou seja, aumentar a produtividade, teve de ser reformulada. Hoje, a proposta da gerência estratégica da produção consiste em administrar o sistema produtivo de modo a poder realizar as estratégias competitivas estabelecidas. O pensamento fundamental que está por trás desta nova filosofia é "a produção afeta a estratégia e a estratégia afeta a produção". [SKINNER, 1985]

Apesar de sua alta eficiência, os sistemas automatizados de produção em massa, tais como as linhas de produção, não são adequados a produções intermediárias, tanto no volume quanto na variedade de produtos, tornando-se deficitários, quando a demanda exige mudanças nas tarefas de manufatura, devido a sua inflexibilidade.

Por outro lado, os tradicionais sistemas de produção por lote permitem alta flexibilidade de fabricação em termos de variabilidade de produtos e volume. No entanto, costumam lidar com altos estoques de produtos em processos e uma baixa taxa de utilização do maquinário. Alguns estudos estimam que neste tipo de produção, uma peça consome 95% de seu tempo esperando ou movendo-se entre máquinas [GREENWOOD, 1988].

A indústria de manufatura, especialmente a indústria metal-mecânica, está movendo-se do ambiente de produção em massa para a produção de pequenos lotes.

A adaptação às novas tecnologias e às novas condições de mercado tem levado as empresas a considerar a adoção de uma terceira possibilidade em termos de automação, os chamados Sistemas Flexíveis de Manufatura (SFM). Tais sistemas caracterizam uma nova era na fabricação discreta de peças.

Os Sistemas Flexíveis de Manufatura são projetados para impor a eficiência das linhas de produção à fabricação em lotes, melhorando sua produtividade pelo aumento da taxa de utilização das máquinas, redução no tempo do ciclo de produção e redução nos estoques de produtos em processo.

O projeto de um SFM requer uma abordagem sistemática e integrada. Browne, Chan e Rathmill [BROWNE et alii., 1984] e WAHAB [1986], identificaram três fases de atividades no desenvolvimento destes sistemas: fase de planejamento, fase de detalhamento do projeto físico e fase de implementação, que inclui o planejamento e o controle operacional do sistema.

Na fase de planejamento a gerência de produção preocupa-se com o que vai ser fabricado, como será fabricado e em que quantidades. É nesta fase também que são definidos objetivos financeiros e estratégicos.

A fase do projeto físico preocupa-se com a seleção de equipamentos, os quais, associados a um conjunto de estratégias e objetivos operacionais, deverão guiar os projetistas na busca de um SFM apropriado, de tal forma, que sejam alcançados os objetivos identificados na fase de planejamento. É também dentro desta fase, que ocorre uma das decisões mais importantes: aquela que trata da definição dos tipos e graus de flexibilidade necessários ao sistema, de tal forma, que possam ser satisfeitos os objetivos estratégicos estabelecidos na fase anterior.

Finalmente, a fase de implementação, relaciona-se com a posta em marcha dos equipamentos, do hardware e software, com considerações sobre o arranjo físico e, principalmente, a definição e detalhamento das estratégias de controle operacional do sistema.

Este trabalho de pesquisa trata de alguns dos mais importantes aspectos referentes à terceira fase do desenvolvimento de um projeto de SFM.

1.1. Origem do Trabalho

O planejamento e controle operacional de um SFM é uma tarefa considerada complexa. Tal complexidade decorre, inicialmente, do fato destes sistemas envolverem muitos elementos conectados entre si, tais como, centros de usinagem, máquinas tipo CNC ou DNC, sistemas automáticos de transporte e manuseio de materiais, sistemas automáticos para manuseio e troca de ferramentas, além de um amplo sistema computacional para controle das operações.

Outro motivo da complexidade de se planejar e operar estes sistemas, é que seu gerenciamento, deve estar intimamente relacionado com as estratégias competitivas da empresa. [Em geral, a decisão de instalação de sistemas de manufatura com tal grau de automação e flexibilidade, decorre da necessidade de imprimir um ritmo mais dinâmico à

função de manufatura, com uma necessária redução em todo o ciclo de produção e um aumento na capacidade de produzir uma maior variedade de componentes e produtos.]

Devido portanto a esta sua natureza de funcionamento integral, a utilização apropriada e eficiente de todos os componentes dos SFM depende, sobretudo, da qualidade das decisões tomadas durante as fases de planejamento e de controle operacional. Decisões errôneas e ineficientes levam, seguramente, à baixa utilização de máquinas, a longos ciclos de produção e a congestionamentos, com conseqüente sub-utilização de suas vantagens estratégicas proporcionadas pela sua potencial flexibilidade.

A origem e motivação deste trabalho baseia-se em alguns fatos:

- no crescente interesse por sistemas de produção com alto grau de automatização, como os SFM;
- na escassez de modelos e programas computacionais voltados ao tratamento de problemas associados as fases de planejamento e controle operacional destes sistemas e,
- no reduzido número de ferramentas computacionais que permitam à avaliação do desempenho operacional dos SFM.

O problema identificado e detalhado a seguir trata da falta de uma metodologia apropriada, que permita avaliar, com facilidade, as inter-relações entre a fase de planejamento e a fase de controle operacional de SFM, diante das inúmeras alternativas e estratégias a elas associadas.

1.2. Definição do Problema

Com base nos motivos que nos levaram ao desenvolvimento deste trabalho, podemos definir dois problemas principais a serem aqui tratados. O primeiro deles, é decorrente da dissociação entre a fase de planejamento e a fase de operação dos SFM, com conseqüentes descompassos e a geração de soluções pobres e ineficientes. O segundo problema, é da pouca disponibilidade de sistemas computacionais que incorporem modelos voltados para o encaminhamento de soluções do problema acima citado e, ao mesmo tempo, sejam suficientemente inteligentes e amigáveis para auxiliar o decisor ou analista, nesta complexa tarefa.

O primeiro dos problemas citados, isto é, o descompasso entre as fases de planejamento e operação, é comum em praticamente todo o tipo de sistema produtivo. Este descompasso, no entanto, costuma agravar-se, na medida em que crescem as opções disponíveis para o decisor. Obviamente, nos SFM, com seu alto grau de flexibilidade, o número de possíveis estados do sistema, é grande, crescendo, portanto, as dificuldades de compatibilização entre as duas fases.

Entre as decisões afetas à fase de planejamento de curto prazo no início de cada ciclo produtivo, a principal delas, diz respeito a definição sobre quais máquinas do sistema deverão realizar as operações necessárias em cada uma das peças a serem manufaturadas. Esta decisão, é tomada com base na capacidade produtiva de cada um dos recursos e no planejamento dos processos de cada uma das peças. Um bom plano para a fase operacional, deve considerar também, a disponibilidade e alocação de ferramentas, além da capacidade do sistema de transporte. Precedem estas decisões, a determinação do mix e do volume de peças ou componentes a serem manufaturados, além das estimativas de duração do ciclo de produção e, as taxas desejadas de utilização das máquinas e dos veículos transportadores.

É comum, nesta fase, a utilização de modelos matemáticos, como a programação linear (PL), por exemplo, para realizar a alocação ou designação dos recursos disponíveis, considerando as várias restrições impostas. No entanto, devido ao fato de modelos como estes não considerarem toda a dinâmica associada à operação dos sistemas, muitas vezes ocorrem conflitos entre o planejado e o que, de fato, pode ser realizado.

Já no nível operacional, o problema pode ser definido como: dado um conjunto de soluções para a carga das máquinas e um conjunto de peças a serem fabricadas, como deverão estas peças serem introduzidas no sistema para serem manufaturadas, qual deve ser a ordem de prioridades nas filas de máquinas e transportadores e, como o sistema de transporte deve ser selecionado e despachado, de tal forma que se possa alcançar as metas de produção, considerando algum tipo de medida de desempenho. Além destas decisões, é necessário definir ainda, alguns números relativos aos parâmetros secundários tais como o tamanho dos buffers e o número de pallets.

Muitas destas decisões, são tomadas durante a própria operação do sistema. isto é, em tempo real. Por exemplo, devido à finita capacidade dos buffers e à limitação imposta no

número de pallets, o número de peças circulando pelo sistema, em um dado momento, é também limitada. Portanto, com o objetivo de assegurar a performance do sistema, é necessário decidir a ordem de entrada das peças.

Alguns dos parâmetros utilizados durante a fase de operação do sistema, tais como os citados número de pallets, tamanho de buffers e regras de despacho, influenciam fortemente o desempenho operacional deste. As decisões tomadas no planejamento, no entanto, são independentes destes fatores. Como consequência, existe sempre uma alta possibilidade de o que foi planejado não ser viável ao nível operacional, a menos que, se possa de alguma forma incorporar as decisões relativas a uma fase, na outra.

O segundo problema a ser tratado neste trabalho, diz respeito a forma de encaminhamento de soluções aos problemas criados pela incompatibilidade entre as fases de planejamento e operação dos SFM. Há uma carência de ferramentas adequadas à avaliação de alternativas de solução para o problema.

As ferramentas existentes, ou não são plenamente adequadas, ou pecam pela dificuldade de sua utilização. Além disso, poucas possuem a capacidade de guardar, pelos menos em parte, a experiência adquirida pelo analista ao longo dos inúmeros experimentos que devem ser realizados na busca de soluções adequadas.

1.3. Objetivos do Trabalho

O objetivo principal deste trabalho é desenvolver e implementar um modelo que permita a avaliação do desempenho operacional de um SFM, considerando as inter-relações entre as fases de planejamento e de controle operacional.

O modelo fará uso de um programa de simulação que permita representar todas as particularidades da fase operacional dos SFM. Além disso, dado que a principal contribuição dos modelos de planejamento à fase operacional é a alocação “a priori” das máquinas as várias seqüências de operações de peças, será incorporado um algoritmo ao modelo, que permite a alocação dinâmica dos recursos, de acordo com o estado do sistema, no momento da decisão.

O que se espera com a retirada desta atribuição da fase de planejamento e sua incorporação a fase de operação, é considerar, dentro da última, uma das mais importantes atribuições da primeira. Uma vez que durante a operação é possível monitorar, por meio de algumas variáveis, o estado do sistema, acreditamos que um algoritmo, que se baseie nestas variáveis, possa ter um desempenho equivalente aquele dos modelos de planejamento, com a vantagem de fazê-lo na própria fase de operação.

No plano real, um algoritmo desta natureza, deve desempenhar um papel equivalente a um sistema computacional controlador do estado e das operações nos SFM. Como veremos, estes sistemas de manufatura possuem, dentre outras características, um sofisticado sistema computacional hierárquico para o controle de todas as ações ali executadas.

Para permitir a utilização plena do modelo é, também objetivo deste trabalho, o desenvolvimento de um protótipo computacional do mesmo. Esta ferramenta deverá incorporar um módulo de simulação de SFM, um módulo inteligente para a interpretação e análise dos resultados das simulações e uma interface gráfica com o usuário. O desenvolvimento e a incorporação de cada um destes módulos ao protótipo tem objetivos específicos que, somados, deverão contribuir para uma melhor compreensão e aplicação da metodologia.

O módulo de simulação tem como objetivo específico, permitir, a avaliação do desempenho operacional do SFM, diante de diferentes estratégias de planejamento e controle operacional da produção.

Quanto ao módulo inteligente para a interpretação e análise de resultados, seu desenvolvimento visa incorporar o conhecimento e a experiência do engenheiro de produção, necessários ao processo de avaliar o desempenho do SFM, quando se considera o planejamento e a operação do processo produtivo em função dos objetivos maiores de compatibilização entre produção e estratégia.

A incorporação se faz através do uso de técnicas de inteligência artificial, mais especificamente, por meio de um sistema especialista. Pelo uso destas técnicas, este estudo pretende que possamos alcançar uma melhor compreensão das principais ações e atitudes, em outras palavras, regras, usadas pelos analistas e engenheiros de produção, na tarefa de melhorar o desempenho de um projeto a partir de resultados obtidos em análises

anteriormente realizadas. Muitas destas regras, fazem parte da arte da simulação e da engenharia de produção, que se desenvolvem na medida em que, os que as praticam, ganham mais experiência.

Finalmente, a incorporação de uma interface gráfica, tem o objetivo explícito de permitir ao usuário uma compreensão plena de todos os processos em andamento, de tal forma que este possa maximizar a utilização do sistema.

1.4. Importância do Trabalho

Decisões relativas à implantação de um SFM costumam, na maioria das vezes, ter impacto direto sobre os custos e a qualidade dos produtos e, portanto, contribuem muito na determinação do papel da empresa, em seus vários mercados, com relação a clientes e competidores.

Conforme citado no item 1.1, "Origem do Trabalho", a etapa de colocar em marcha um projeto de SFM é considerada uma tarefa complexa. Esta complexidade decorre tanto das decisões ao nível operacional, quanto daquelas ao nível do planejamento estratégico da produção, que se caracterizam por uma diversidade de combinações de políticas e parâmetros operacionais, com implicações diretas na estratégia competitiva da empresa.

Com base nestas complexidades, verificamos a importância do desenvolvimento de metodologias e ferramentas que possam auxiliar os decisores nesta importante etapa dos projetos de SFM.

A posse de um modelo e de um sistema computacional, que permitam ao decisor testar e avaliar várias alternativas para o planejamento e controle da produção de projetos de SFM, considerando, não somente as diversas combinações de equipamentos e estratégias operacionais, mas, também, as implicações destas sobre o desempenho estratégico do sistema testado, certamente contribuirá para minimizar os esforços associados a estas tarefas.

A importância desta pesquisa, está, portanto, no fato dela proporcionar o desenvolvimento deste modelo, em conjunto com um protótipo de uma ferramenta computacional voltada para avaliar o desempenho operacional de SFM, considerando a complexidade desta

tarefa. Além disso, é conveniente ressaltar ainda, sua contribuição ao desenvolvimento dos chamados sistemas inteligentes de simulação, segmento que vem assumindo mais e mais um importante papel, principalmente nas aplicações voltadas a simulação de sistemas automatizados de produção.

1.5. Suposições, Alcance e Limitações do Trabalho

Algumas limitações e suposições iniciais são necessárias de forma a tornar tratáveis os problemas envolvidos nos objetivos definidos neste trabalho, tanto do ponto de vista teórico como prático.

Todo SFM é único e dependente de sua aplicação, da variedade de peças a serem produzidas e da flexibilidade caracterizada pelos seus componentes (hardware e software). De acordo com este pressuposto, torna-se quase impossível a criação de um modelo e conseqüente ferramenta computacional, que permita a avaliação de todo e qualquer SFM. Desta forma, torna-se imperativo definir o ambiente físico considerado nesta pesquisa.

Este trabalho está voltado para o segmento da indústria metal-mecânica, produtora de peças por processos não-contínuos. Consideramos aqui aquele tipo de ambiente de manufatura no qual é produzida uma variedade razoável de peças, simultaneamente e com demanda limitada, isto é, um ambiente típico para os SFM.

É suposto que o capital investido no sistema é economicamente justificado e que todas as etapas preliminares de decisão de instalação do sistema já tenham sido levadas a efeito.

Sistemas com "n" máquinas são considerados. Estas máquinas ou centros de usinagem, não são necessariamente idênticas. Para as tarefas de transporte e manuseio de peças e materiais são considerados os chamados veículos-auto-guiados ou VAGs.

Assumimos que os centros de usinagem e o sistema de transporte e manuseio de materiais (STMM) são controlados por um computador central à célula de manufatura e que todos os programas para a produção ou transporte de peças podem ser carregados sem nenhum tipo de espera. As máquinas podem possuir ou não um local próprio para espera/armazenagem (buffer) de produtos em processo. Os sistemas devem possuir uma

estação automática para carregamento/descarregamento de peças e sua capacidade não é considerada uma restrição.

O problema da alocação de ferramentas não é tratado neste trabalho. Assumimos que o planejamento de sua utilização, alocação e armazenagem já foi executado e que qualquer máquina possui disponibilidade imediata da ferramenta necessária.

Assumimos, também, que o arranjo físico e os caminhos dos VAGs, são conhecidos e que a distância entre duas máquinas quaisquer é fixa. São também conhecidos o mix e a demanda de peças em cada cenário simulado.

Para que o processo de simulação e análise seja iniciado, assumimos que um modelo inicial, operacionalmente viável, seja fornecido.

As análises efetuadas estão limitadas aos conhecimentos incorporados ao sistema especialista, os quais, poderão ser ampliados, pela incorporação de novos conhecimentos relacionados com a seleção de regras associadas ao planejamento e controle da produção, objetivando a avaliação do desempenho operacional e/ou estratégico de SFM. Outros tipos de análises ou avaliações, tais como econômica ou financeira, não são aqui abordados.

Portanto, o alcance deste trabalho, limita-se à determinação e avaliação dinâmica do desempenho operacional dos SFM definidos ou propostos, bem como de sua capacidade de adaptação diante diferentes ambientes, considerando-se diversas estratégias de planejamento e controle operacional.

1.6. Organização do Trabalho

Este trabalho divide-se em sete capítulos. Neste primeiro, comentamos sobre os motivos que o originaram com uma conseqüente definição dos problemas a serem tratados e dos objetivos da pesquisa. Comentamos, também, sobre sua importância e suas limitações. No segundo capítulo, apresentamos um levantamento do estado da arte dos diversos assuntos diretamente tratados ou correlacionados com a pesquisa. No terceiro capítulo, desenvolvemos o modelo de simulação proposto, acompanhado do algoritmo para a alocação dinâmica dos recursos. Segue-se um capítulo no qual realizamos uma série de experimentos para testar-mos e compararmos os resultados do modelo desenvolvido. O

quinto capítulo é dedicado ao desenvolvimento do protótipo computacional. Nele apresentamos detalhes de sua construção, que inclui os módulos para a simulação e análise de SFM e da interface gráfica com o usuário. No sexto capítulo, apresentamos uma demonstração do uso do protótipo. Fazemos uso do mesmo sobre um problema de avaliação do desempenho operacional de um SFM, sob algumas combinações de parâmetros e políticas operacionais. No sétimo e último capítulo, realizamos um apanhado geral de todo o trabalho de pesquisa e experimentação desenvolvido, salientando suas contribuições e suas limitações. Encerramos o trabalho com algumas sugestões a futuras pesquisas.

CAPÍTULO II

REVISÃO BIBLIOGRÁFICA

Esta revisão bibliográfica aborda sete temas principais. Primeiramente, discute-se o conceito de flexibilidade e sua importância estratégica a nível de sistemas de produção. Na sequência discute-se os sistemas flexíveis de manufatura, apresentando-se conceitos, definições e seus principais componentes. Uma vez que o modelo a ser desenvolvido neste trabalho envolve os problemas e decisões afetas ao planejamento e controle operacional de SFM, a terceira seção desta revisão é dedicada a este assunto.

O quarto tema tratado é o da avaliação de SFM. Apresenta-se uma revisão das diversas abordagens propostas para a avaliação destes sistemas, bem como dos instrumentos utilizados nestas avaliações. O quinto assunto discutido trata de modelos de simulação de SFM, técnica utilizada na metodologia desenvolvida para avaliação de desempenho de projetos destes sistemas. Tendo em vista suas relações com a metodologia e o modelo desenvolvido neste trabalho, o sexto tema abordado, diz respeito a sistemas especialistas. São apresentadas algumas definições, conceitos básicos, bem como, algumas das ferramentas e metodologias existentes para seu desenvolvimento. Discute-se, na última seção, algumas relações, similaridades e formas pelas quais podem ser combinadas as técnicas de simulação e sistemas especialistas, para auxiliar nos processos de decisão. Os resultados destas combinações, são os chamados sistemas inteligentes de simulação. Sua definição e uma discussão de seu uso na avaliação de SFM, também são tratados.

2.1 Flexibilidade e Sistemas de Produção

Flexibilidade é, literalmente definida como, a capacidade de dobrar-se sem quebrar, isto é, adaptar-se [GERWIN, 1982].

Com o advento das novas tecnologias baseadas em microprocessadores, o conceito de flexibilidade reaparece como um elemento chave no projeto, operação e gerenciamento de

sistemas de produção. Uma grande quantidade de literatura sobre flexibilidade aplicada à produção foi acumulada nos últimos dez anos, conforme levantamentos de Gupta e Goyal [GUPTA, 1989] e de SETHI [1990]. A maior parte dela dedicada à definição dos vários tipos de flexibilidade e à identificação de sistemas que exibem ou não um ou mais destes tipos. Alguns artigos dedicam-se também às formas de medi-las. Fica claro da literatura levantada que flexibilidade é um conceito complexo, multidimensional, e não muito fácil de ser entendido.

Segundo SETHI [1990], pelo menos 50 diferentes termos para vários tipos de flexibilidade podem ser encontrados na literatura sobre sistemas de produção, muitas vezes referindo-se ao mesmo tipo de flexibilidade.

Percebe-se também que pouco esforço tem sido feito, tanto no sentido de um tratamento analítico, quanto na determinação quantitativa destes conceitos. Como resultado, as formas de avaliação propostas, são, na maioria das vezes simples e arbitrárias.

2.1.1 O Conceito e Importância da Flexibilidade na Manufatura

Flexibilidade de um sistema de produção, é sua capacidade de adaptação a um grande número de mudanças [GERWIN, 1982]. JAIKUMAR [1984], enfatiza o fato que a flexibilidade na manufatura é sempre uma restrição dentro de um domínio. Este domínio pode ser definido em termos do portfólio de produtos, processos e procedimentos, podendo ser bem compreendido por projetistas de produtos, engenheiros de produção ou programadores. Este domínio pode ser planejado, administrado e, se bem compreendido, pode ser expandido [GERWIN, 1982; GOLDHAR, 1983].

Com respeito às incertezas ambientais, a flexibilidade na manufatura é necessária para lidar com mudanças internas e forças externas [BUZACOTT, 1985; GARRET, 1986]. Os distúrbios internos incluem a quebra de equipamentos, variações nos tempos das tarefas, esperas em filas, rejeitos e retrabalhos. Forças externas referem-se usualmente as incertezas fundamentais do ambiente de competição. Tais incertezas podem estar presentes na disponibilidade de recursos, no mix e preço dos produtos ou no nível de demanda. Além disso, podem significar mudanças no gosto de consumidores, inovações tecnológicas, novas regulamentações, etc. [GARRET, 1986].

Parece clara portanto a importância da flexibilidade na manufatura no que se refere às forças de uma empresa e seu papel na estratégia de competição destas. Hayes e Wheelwright [HAYES, 1984], consideram flexibilidade como uma das dimensões da estratégia competitiva de um negócio, além de um dos mais notáveis critérios para avaliar-se um sistema de manufatura.

O gerenciamento da flexibilidade na manufatura, invariavelmente traz consigo uma questão: qual o nível ótimo para os vários tipos de flexibilidade? A resposta a esta questão requer que a gerência identifique e esteja apta a medir os vários tipos que a manufatura deve possuir para que possa ganhar vantagem competitiva.

2.1.2. Tipos de Flexibilidades

São vários os tipos de flexibilidades que aparecem na literatura. Muitos esforços foram e são despendidos para classificá-los, quantificá-los e medi-los, sob diferentes aspectos como pode ser verificado em BARAD [1988], BUZACOTT [1982], CHATTERJEE [1984], GUPTA [1989], SETHI [1990] e SLACK [1987].

As definições apresentadas por Browne e outros [BROWNE, 1984], tornaram-se clássicas, aparecendo na maioria dos trabalhos que tratam do tema. Dentre os vários tipos de flexibilidades propostas por estes autores, as quatro citadas abaixo são as mais importantes no que se refere especificamente a sistemas de manufatura.

Flexibilidade de Máquinas: é a habilidade e facilidade para executar as mudanças necessárias à produção de um dado conjunto de tipos de peças, tais como repor ferramentas desgastadas ou quebradas, trocar ferramentas e montar fixadores sem interferência ou longos períodos de tempo.

Flexibilidade de Processo: é a habilidade de variar os passos necessários para completar uma tarefa. Isto permite que diferentes tarefas sejam executadas pelo sistema usando-se uma variedade de máquinas.

Flexibilidade na Variação das Rotas: é a habilidade de variar a seqüência de visitação das máquinas (útil no caso de quebras, por exemplo) e continuar produzindo o dado conjunto de peças. Isto só é possível porém, quando existe mais de uma rota viável para o processo ou quando cada operação pode ser desempenhada por diferentes máquinas.

Flexibilidade de Volume: é a habilidade de operar um SFM, sob diferentes volumes de produção, com lucratividade, considerando-se somente níveis viáveis de produção.

Outros tipos, como por exemplo a flexibilidade operacional de uma peça, a qual se refere a possibilidade de se produzir uma peça de diferentes formas, é considerada uma propriedade da peça a ser produzida e não do sistema que a produz. Existem ainda alguns tipos de flexibilidade que são consideradas decorrentes das demais, isto é, só existem se outras estiverem presentes no sistema. Como exemplo temos a flexibilidade de produção, cuja existência depende, praticamente, de todas as outras, mais a flexibilidade operacional das peças.

2.2 Sistemas Flexíveis de Manufatura

Os sistemas de manufatura, em geral, exibem muitas características comuns, embora diferindo nos seus detalhes. De uma forma ampla, pode-se afirmar que todos compreendem produtos e equipamentos usados para produzi-los, tais como: matérias primas, máquinas, transportadores, armazenadores, ferramentas, operadores, etc.. Suas diferenças refletirão os diferentes modos com que seus integrantes podem ser combinados para formar um particular sistema.

Segundo CARRIE [1988], um sistema significa "um conjunto de coisas ou partes conectadas ou interdependentes, de tal maneira que formem uma unidade complexa".

Em geral, os sistemas de manufatura refletem perfeitamente esta definição, já que são compostos de muitos elementos diferentes, associados, formando uma estrutura organizada e com um objetivo comum.

A literatura costuma apresentar as diferenças básicas entre os diversos sistemas de manufatura, dividindo-os, em dois tipos clássicos: as linhas de produção (transfer lines) e as oficinas de serviços (job shops) [GUPTA, 1989; HAYES, 1984 e HILL, 1985].

Gupta e Goyal [GUPTA, 1989] descrevem bem as diferenças entre os dois grupos:

"...no primeiro, temos as típicas linhas de produção, com máquinas dedicadas, indicadas especialmente para a produção em massa de poucos produtos. Este tipo de processo permite um baixo custo unitário de produção, devido à economia de escala, mas em contra partida apresentam uma baixa flexibilidade.

No segundo grupo, encontram-se aqueles sistemas que usam um conjunto, não integrado, de máquinas de propósito geral, especialmente apropriadas à produção, em pequenos lotes, de uma grande variedade de produtos. Nestes, o custo por unidade produzida tende a ser alto, mas, em compensação, a flexibilidade do processo pode acomodar, com certa facilidade, mudanças de projeto, flutuações de demanda e trocas no mix de produtos".

Mais recentemente, sistemas avançados de manufatura (máquinas do tipo CN ou CNC) associadas a controles informatizados (CAE, CAD, CAM), oferecem uma terceira opção com mais flexibilidade que a produção em linha e menor custo unitário que os obtidos nas produções com máquinas de propósito geral não integradas. A este tipo de sistema de produção a literatura costuma referir-se com o nome de Sistemas Flexíveis de Manufatura ou SFM.

Tais sistemas não possuem uma definição formal e universalmente aceita. GREENWOOD [1988], apresenta o que parece ser uma definição mais geral:

"...é uma cuidadosa combinação de controles computacionais, comunicações, processos de manufatura e equipamentos relacionados, que permitem ao setor produtivo de uma organização responder rápida, economicamente e de maneira integrada às mudanças significativas no seu ambiente operacional. Tais sistemas possuem, tipicamente, equipamentos de

processamento (máquinas-ferramentas, estações de montagem, robôs, etc.), sistemas de manuseio e transporte de materiais (robôs, esteiras, veículos-auto-guiados, etc.), um sistema de comunicações entre os equipamentos e um sofisticado sistema de controle computacional".

Embora esta seja uma definição longa e genérica, ela possui o mérito de não incluir nenhum tipo de máquina ou equipamento específico, algo comum em outras definições, além de cobrir todos os aspectos dos SFM.

2.2.1 Componentes dos Sistemas Flexíveis de Manufatura

Segundo CARRIE [1988] e TALAVAGE [1988], os principais componentes dos SFM são:

- Centros de usinagem numericamente controlados;
- Sistemas automáticos de transporte e manuseio de materiais;
- Sistema computacional para controle em tempo real;

O arranjo físico e os componentes de um SFM típico são mostrados na figura 2.1.

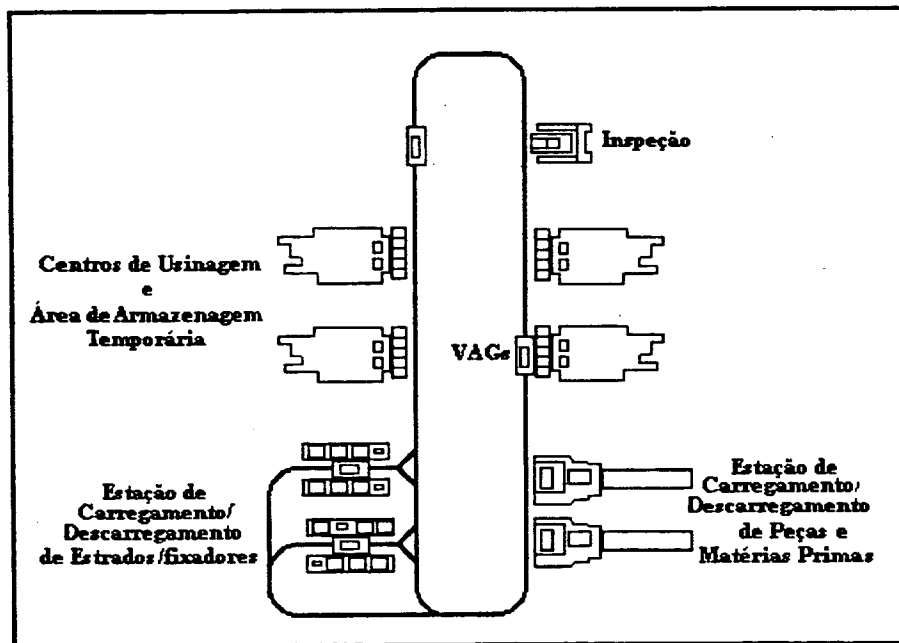


Fig.: 2.1 Um Exemplo de Sistema Flexível de Manufatura

2.2.1.1 Centros de Usinagem Numericamente Controlados

Centros de usinagem numericamente controlados são máquinas-ferramentas multifuncionais, com controle numérico computadorizado (CNC) ou controle numérico direto (DNC), capazes de produzir automaticamente uma grande variedade de peças [WICK, 1987]. O controle numérico independente de múltiplos eixos, geralmente de 4 a 6, fornece flexibilidade a estes centros para executarem uma grande quantidade de operações.

Estes centros de usinagem são, geralmente, equipados com módulos para armazenagem de ferramentas com capacidades que variam entre 40 e 480 ferramentas [WICK, 1987]. Em cada centro de usinagem, com armazenagem própria de ferramentas, encontra-se um extrator automático de ferramentas (EAF), para que trocas rápidas possam ser programadas e efetuadas. O EAF garante uma grande diminuição do tempo de preparação entre operações (setup), aumentando a flexibilidade destes centros de usinagem, com a possibilidade de diferentes operações simultaneamente.

2.2.1.2 Sistemas Automáticos de Transporte e Manuseio de Materiais

Os sistemas de transporte e manuseio de materiais (STMM) são responsáveis pela completa integração dos SFM, fornecendo uma ligação física em tempo real entre seus diversos componentes. Os STMM são necessários para um transporte rápido e seguro de ferramentas, fixadores, peças, matéria-prima, etc., entre as estações de trabalho e entre estas e os setores de armazenagem e carregamento e descarregamento [RADHARAMANAN, 1989; STECKE, 1985]. Uma grande variedade de equipamentos é utilizada pelos STMM, os quais incluem robôs, veículos-auto-guiados (VAGs), esteiras, monotrilhos, etc. [ZISK, 1983].

A seleção de um ou mais destes equipamentos em combinação para a composição de um STMM para um SFM depende das necessidades de manuseio de materiais do sistema. Robôs, por exemplo, são mais usados em operações de carregamento/d Descarregamento de peças e matérias-primas, enquanto esteiras são usadas em transporte de peças e produtos

em processo ao longo de um caminho fixo no sistema. A flexibilidade do transporte por meio das esteiras é muito limitada, por este motivo, quando existe a necessidade de mais flexibilidade de rotas e planos de produção, o uso dos veículos-auto-guiados (VAGs) contribui positivamente para o alcance destes objetivos [CARRIE, 1988; TALAVAGE, 1988 e ZISK, 1983].

De todos os STMM disponíveis, os VAGs são aqueles que proporcionam o mais alto grau de liberdade para a movimentação de peças ou matérias-primas. Além disso, os VAGs possuem muitas outras características desejáveis, tais como flexibilidade de expansão, boa acessibilidade às estações de trabalho, independência do arranjo físico do sistema e uma grande facilidade de interfaciamento com outros recursos auxiliares, tais como, sistemas automáticos de armazenamento ou de carregamento/descarregamento. Por outro lado, algumas das desvantagens são: um investimento de capital relativamente alto, baixa capacidade de carregamento, baixa precisão de posicionamento quando comparados com vagonetas ou monotrilhos.

2.2.1.3 Controle Computacional

A completa integração dos componentes de um SFM, é efetuada de duas maneiras: a interligação física, via STMM e a interligação lógica, efetuada entre computadores e equipamentos, via uma rede local. Como delineado anteriormente, a função básica do STMM em um SFM é transportar peças acabadas ou não em estrados, dos centros de usinagem, ao final de cada operação. Enquanto isso, o objetivo primário dos computadores é controlar e sincronizar todas as funções e operações dos componentes dos SFM. [GROOVER, 1980; TALAVAGE, 1985]

Os principais elementos do sistema de controle são:

- Um computador servidor, cuja tarefa é controlar as atividades dos vários equipamentos dentro do sistema;
- Uma rede de comunicações, cuja tarefa é ligar o servidor as estações de trabalho;
- Finalmente unidades de controle responsáveis pelo detalhamento das atividades nas próprias estações de trabalho - por exemplo, uma unidade de controle

numérico computacional, um microcomputador ou um controlador lógico programável em uma máquina ferramenta ou robô.

Um diagrama de bloco das funções típicas de um controle computacional em um SFM é apresentado na tabela. 2.1..

SFM - Hierarquia e Funções de Controle Computacional	
Nível	Função e tipo de Computador
Fábrica	Integração da Fábrica - Grande Porte
Áreas de Produção	Integração de células de Produção - Minicomputador
Célula/ SFM	Controle à nível de célula - Computador Pequeno Porte
Estações de Trabalho	Controle Individual de Tarefas - Micro ou Controlador Lógico

Tab. 2.1: Hierarquia e funções de controle computacional em um SFM.

A estrutura de controle computacional é hierárquica, com cada nível possuindo funções independentes supervisionadas por computadores, que por sua vez estão submetidos às decisões tomadas nos níveis superiores. Na base desta pirâmide, equipamentos individuais, tais como, centros de usinagem, STMM, sistemas de armazenamento e trocas de ferramentas, etc., são controlados por micros industriais ou controladores lógicos programáveis. Aí, decisões sobre processamento, transporte e armazenagem de peças, alocação e transporte de ferramentas são passadas de um nível superior aos controladores individuais que se encarregam então de executar estas tarefas conforme uma programação previamente carregada.

Já no nível imediatamente superior, isto é, das células, são tomadas as decisões sobre a alocação dos recursos, tamanho de lotes, despachos e sequenciamentos, envolvendo todos os elementos da célula, conforme as ordens superiores, encaminhadas pelo nível mais alto no controle hierárquico, com decisões de curto e médio prazos que abrangem toda a área de produção.

2.3 Planejamento e Controle Operacional de SFM

O planejamento e o controle da operação de SFM tem sido muito investigado nos últimos anos. Entre as inúmeras abordagens, consideramos mais apropriado apresentar nesta revisão aquelas que tratam de modelos de planejamento da produção, especialmente os que se referem a alocação dos recursos, e de modelos voltados ao tratamento de problemas da fase operacional dos SFM. Consideramos, também, alguns trabalhos que tratam das inter-relações entre as duas fases.

2.3.1 Planejamento da Operação

A operação de um SFM requer uma cuidadosa preparação do sistema antes de sua operação. Vários problemas relacionados a fase de planejamento devem ser equacionados para que se possa obter uma utilização eficiente do sistema. Uma das mais importantes tarefas do planejamento é, sem dúvida, a designação sobre onde, isto é, em qual máquina, será realizada cada uma das operações necessárias a manufatura de cada tipo de peça. Várias abordagens são utilizadas. As mais comuns são a programação linear inteira e a programação com múltiplos objetivos. Aparecem, também, trabalhos envolvendo programação dinâmica e teoria dos grafos.

STECKE [1986] propôs uma abordagem hierárquica. O problema era dividido em dois níveis, agregado e desagregado. No primeiro, uma solução ótima para as taxas de produção e agrupamentos de máquinas era obtida. No segundo, modelos de programação linear inteira são utilizados para gerar a designação de peças à máquinas.

Com o objetivo de minimizar os custos de processamento, KUSIAK [1985] modelou o problema como uma programação linear inteira do tipo 0-1. Os tempos de processamento são sempre considerados iguais e, são desconsiderados, sobre o sistema, os impactos de congestionamentos e da capacidade do sistema de transportes.

Leung e Tanchoco [LEUNG, 1987], desenvolveram um modelo de programação inteira para o planejamento, em um sistema voltado ao estudo de reposição de equipamentos em sistema integrados. Suas investigações no ambiente de SFM, faz algumas considerações a capacidade do sistema de transportes de peças e materiais

Leung e outros [LEUNG, 1990] desenvolveram outro modelo de planejamento de SFM, usando programação linear inteira, considerando explicitamente as restrições impostas pelo sistema de transportes, o qual aparece como responsável pela movimentação de peças e materiais nas e entre as estações de trabalho e, entre estas e uma estação de carregamento e descarregamento. Resultados do modelo demonstram a significância da capacidade e do custo do sistema de transporte sobre os resultados do planejamento da produção.

Num trabalho recente e bastante abrangente, EGBELU [1993], propõe uma solução para o problema do planejamento de SFM, usando duas abordagens: programação dinâmica e teoria dos grafos. A chave de seu processo de otimização passa pela flexibilidade de rotas daqueles sistemas. Sua abordagem propõe a integração de duas funções básicas nos problemas da manufatura: o planejamento do processo, especificamente no que trata do sequenciamento das operações e no controle dos estoques intermediários, de fundamental importância nos SFM com tamanhos limitados de buffers. Custos de setup e processamento e tamanhos de lotes são considerados na busca da minimização do custo total.

O planejamento dos processos em SFM requer o alcance de vários objetivos, os quais são conflitantes por natureza. É comum, pois, a presença na literatura de alguns trabalhos reportando a utilização de técnicas que procurem balancear o compromisso entre os múltiplos objetivos. Goal programming é uma das técnicas empregadas.

Utilizando-se desta técnica, O'Grady e Menon [O'GRADY, 1984] desenvolveram um modelo de planejamento para SFM. A necessidade total de produção das peças foi dividida em grupos de acordo com as datas de entrega, sendo otimizadas sob este aspecto. A capacidade das máquinas e dos magazines de ferramentas também foram consideradas.

Outro modelo de planejamento usando esta técnica foi apresentado por Lee e Jung [LEE, 1989]. Os vários objetivos definidos são o volume de produção de cada tipo de peça, o balanceamento da carga das máquinas, a capacidade de fixadores e pallets e o tempo de processamento. Entretanto, o trabalho não sugere nenhum procedimento para determinar metas para cada um dos objetivos.

2.3.2 Controle da Operação

Ao nível do planejamento, vários dos aspectos operacionais do sistema não são capturados. Entretanto, durante a fase de execução, tais aspectos possuem significantes impactos sobre o desempenho do sistema. Entre estas considerações, algumas referem-se a decisões que devem ser tomadas em tempo real, tais como o momento de liberar uma peça ao sistema, o tempo de iniciar uma operação ou de transportar uma peça. Outras decisões precisam ser geradas em função de eventos inesperados, tais como a falha de uma máquina e a quebra de uma ferramenta ou transportador.

A modelagem matemática de problemas de controle operacional, envolvem um grande número de variáveis inteiras, dificultando sua solução. Mesmo problemas envolvendo sistemas com um único centro de usinagem, como o proposto e tratado por Raman e outros [RAMAN, 1989], apresentam grandes dificuldades de solução, a menos que uma série de restrições sejam previamente feitas. Daí, a importância das decisões tomadas ao nível do planejamento, antes do encaminhamento das decisões referentes ao controle operacional.

Nas subseções a seguir, revisamos alguns dos problemas afetos às decisões da fase operacional, tais como despacho de peças e veículos, além de suas interações com alguns dos principais parâmetros dos SFM, isto é, pallets, buffers, etc..

2.3.2.1 Despacho de Peças

O despacho de peças trata de dois problemas: o do gerenciamento ou ordenação da entrada de peças no sistema e o da ordenação ou priorização de peças que encontram-se nos buffers das máquinas.

Os problemas relativos a ordem de entrada das peças no sistema foram bem tratados em um estudo no qual Carrie e Petsopoulos [CARRIE, 1985] utilizaram-se de um modelo de simulação. Neste estudo, sete diferentes regras de gerenciamento da entrada das peças foram testadas. Um experimento foi projetado para analisar a significância destas regras para com a performance do sistema. Os resultados, embora significativos, foram

dependentes dos tipos de peças uma vez que as regras utilizadas estavam baseadas nas cargas de trabalho de cada tipo de peça.

Vários outros autores citam trabalhos de investigações sobre a importância das regras para entrada das peças.[ARBEL, 1984; MONTAZERI, 1990]. Seguindo a mesma linha investigatória, MAHESHWARI [1992] realizou um trabalho, o qual consideramos um dos mais importantes. Em sua pesquisa, ele realizou exaustivos experimentos com várias regras de despacho, de forma individual ou combinada com regras de sequenciamento de peças em filas e de despacho de veículos e parâmetros operacionais. Suas conclusões, demonstram a importância destas regras sobre o desempenho operacional dos SFM.

Com relação ao segundo tipo de problema associado ao despacho de peças, isto é, a ordenação ou priorização de peças que se encontrem em filas ou buffers de máquinas, o número de trabalhos na literatura é bastante amplo. Alguns destes trabalhos merecem atenção. Nof e outros [NOF, 1979] reportam, em uma das primeiras pesquisas sobre este assunto voltada a células automatizadas de manufatura, a significância desta ordenação. Quatro diferentes regras foram utilizadas: FIFO, Menor Tempo de Processamento Restante (MTPR), Menor Tempo de Processamento (MTP) e Maior Tempo de Processamento. Os resultados apontaram para a significância das regras mas, ao mesmo tempo, citam que esta depende do mix, do tamanho dos lotes e da participação relativa de cada tipo de peça.

Stecke e Solberg [STECKE, 1981] modelaram um SFM da Caterpillar em Peoria, Illinois, EUA. O sistema composto de nove máquinas, tem um número de pallets e fixadores restritos. Dezesesseis regras de ordenação em conjunto com seis políticas de carregamento foram testadas. A combinação Menor Taxa de Produção (MTxP) para carregamento e Tempo Total de Processamento obteve o melhor desempenho. Os autores no entanto citam que este desempenho depende extraordinariamente dos parâmetros do sistema.

Outro trabalho interessante foi realizado por Wilhelm e Shin em 1985 [WILHELM, 1985]. Os autores desenvolveram uma ordenação heurística, considerando um SFM com armazenagem centralizada de peças em processo. A heurística considera rotas alternativas e pode designa-las dinamicamente. A prioridade é determinada sobre peças que se

encontrem no buffer central. Eles reportam ganhos de desempenho do sistema em relação a processamentos sem a heurística.

Com a hipótese de que uma combinação de regras obtém melhores resultados em um SFM do que uma única regra, Wu e Wysk [WU. 1989] desenvolveram um modelo de simulação para verificar qual combinação obterá os melhores resultados sobre um determinado sistema. O experimento foi realizado considerando diversos períodos de tempo, aos quais os autores denominaram “janelas de simulação”. Ao final de cada período, os resultados da simulação eram examinados e a melhor regra, com base no estado do sistema, era então escolhida para ser utilizada no período seguinte.

Em outro trabalho cujas decisões são tomadas com base no estado do sistema, ZEESTRATEM [1990], desenvolveu um procedimento para a programação de SFM. O estado do sistema é baseado em um banco de dados e é constantemente atualizado. A convergência do procedimento não é discutida. O autor compara seu procedimento com dez diferentes regras de ordenação. Os resultados mostram que seu procedimento obtém os melhores resultados. Além disso é bastante consistente em termos da variância medida sobre o desempenho do sistema.

No mesmo trabalho investigatório já anteriormente citado, MAHESHWARI [1992], desenvolveu uma extensa pesquisa que resultou na elaboração de um modelo envolvendo tanto a fase de planejamento (com programação linear) como a fase de operação com simulação. Na fase de operação, diferentes regras para o controle operacional foram testadas, em combinação com quatro modelos de planejamento e diferentes parâmetros operacionais. Os resultados apontam para diferentes combinações proporcionando melhor desempenho do sistema. De maneira geral, as combinações que obtiveram os melhores resultados envolveram a regra MTP para o controle das filas, sendo esta considerada a melhor regra quando se tratava de despacho de peças em filas.

2.3.2.2 Despacho de Veículos

Torna-se evidente das discussões sobre planejamento da produção de SFM que uma peça deve realizar uma sequência de visitas aos diversos centros de usinagem, o que depende da ordem de suas operações e das capacidades operacionais destes centros além,

é claro, das restrições tecnológicas impostas. O transporte destas peças entre os diversos centros, ao final de cada uma das operações, é que gera o problema do despacho de veículos. As decisões tomadas sobre este problema, possuem considerável impacto sobre o desempenho de sistemas integrados de manufatura, como pode ser verificado nos trabalhos que citamos a seguir.

Em 1984, Egbelu e Tanchoco [EGBELU, 1984], classificaram as tarefas de despacho de veículos em duas categorias. A primeira envolve a seleção de um veículo vazio, por uma peça, quando esta termina uma operação e necessita de transporte. A segunda categoria, envolve a seleção de uma peça para transporte, entre as várias que estejam esperando por transporte.

Em um procedimento sugerido por Raman e outros [RAMAN, 1989] o transporte das peças era considerado como uma operação e o tempo de transporte como o tempo de uma operação. Assim, as decisões de planejamento envolviam, simplificadaamente apenas operações, no seu sentido mais amplo. As prioridades eram estabelecidas de acordo com algumas regras baseadas no cumprimento de prazos de entrega. Os resultados apontam alguns conflitos no uso dos veículos.

Um trabalho importante foi elaborado por Bozer [1989]. Neste estudo foram analisados regras e procedimentos existentes para o despacho de veículos. Bozer ressalta a necessidade de um método consistente para a avaliação do desempenho do sistema para se poder verificar a significância de cada uma das regras, individualmente ou combinadas. A maioria dos testes foram realizados sob condições específicas, não podendo, portanto serem generalizadas. Importante neste trabalho, no entanto, foi esclarecer o fato de que regras para despacho de veículos não devem ser testadas independentemente de outras decisões relacionadas com o planejamento e controle de operações em SFM, tais como despacho de peças, ordenamento de peças ou alocação de máquinas.

MAHESHWARI [1992] também incluiu análises sobre a influência das regras de despacho de veículos sobre o desempenho dos SFM. Suas conclusões apontam para uma regra baseada no Mínima Carga de Trabalho na Fila da Próxima Máquina (MCTPM), isto é, da fila de peças esperando por transporte junto aos buffers de saída das máquinas, é

escolhida a peça, cuja máquina que realizará sua operação seguinte, apresentar a menor carga de trabalho.

Dois outros trabalhos retratam também a importância das regras de despacho de veículos. L. Zhuang e Hindi [ZHUANG, 1993], avaliaram um SFM adotando o lead time das peças como medida o desempenho. Usando um modelo baseado em redes de filas, estudaram as inter-relações entre regras de despacho e flexibilidade de rotas dentro de SFM e seu impacto sobre o desempenho operacional do sistema.

Outro trabalho a destacar foi o realizado por Yim e Linn [YIM, 1993]. Neste estudo, os autores trataram várias regras de despacho de VAG's, além de outros parâmetros, sobre SFM puxados e empurrados. O sistema foi modelado e simulado usando redes de Petri para investigar os efeitos das diversas regras e parâmetros investigativos sobre o modelo. Os autores concluem, que não existem diferenças significantes sobre o desempenho do sistema entre as regras de despacho de veículos, adotando a política de puxar ou empurrar, quando os SFM trabalham com uma alta taxa de ocupação.

2.3.2.3 Impacto dos Parâmetros do Sistema

No planejamento, é difícil formular restrições impostas pelo sistema, com base nos recursos logísticos e demais características tecnológicas do sistema. Estes recursos e características podem incluir o número de VAGs, o número de pallets, o layout do sistema ou o tamanho dos buffers. Outras características podem ser a relação entre o número de pallets e buffers ou pallets e número de máquinas. Estes parâmetros, possuem, definitivamente impacto sobre as características operacionais dos SFM.

Em um estudo sobre sistemas com tamanhos limitados de buffers, CO e JAW [CO, 1988], não encontraram muitas evidências sobre a importância das regras de ordenação de peças quando os buffers eram pequenos. Isto é, a utilização de diferentes regras de ordenação não produzia impacto sobre a performance do sistema quando o número de peças a serem ordenadas nos buffers era limitado. No entanto, na medida em que crescem os tamanhos de buffers, crescem os estoques intermediários e a importância de seu gerenciamento

sobre o desempenho dos sistemas. Este impacto foi verificado pelos autores quando do aumento dos buffers e do número de pallets.

Outros trabalhos importantes foram os de Schriber e Stecke [SCHRIBER, 1988] e o de MAHESHWARI [1992], demonstrando o impacto da variação dos parâmetros secundários do sistema sobre a taxa de utilização das máquinas e transportadores e, principalmente sobre o desempenho operacional do sistema. Os seus resultados demonstraram, por exemplo que um aumento nos buffers ou no número de VAGs nem sempre melhoram a performance. Em alguns casos, esta até decai em função de congestionamentos no sistema.

2.4 Avaliação de Projetos de Sistemas Flexíveis de Manufatura

Durante o ciclo de vida de um SFM são várias as fases que exigem decisões sobre o sistema. Dentre estas pode-se citar: planejamento do sistema, projeto inicial, projeto detalhado, instalação, planejamento e programação da produção, operação e eventuais modificações e/ou melhorias efetuadas após sua posta em marcha [SURI, 1985; TALAVAGE, 1988]. As decisões associadas a cada uma destas fases exigem constantes avaliações do projeto, as quais, para serem executadas, necessitam de medidas de desempenho e de modelos de avaliação apropriados.

As medidas de desempenho são definidas de acordo com os objetivos e metas do decisor e estabelecem os padrões de desempenho desejados.

A definição sobre os modelos é mais ampla e é consequência, tanto das medidas de desempenho adotadas como do tipo de avaliação desejada. A avaliação e, por consequência, o modelo a ser adotado, deve considerar, principalmente, a complexidade das decisões que serão tomadas.

A metodologia e a ferramenta desenvolvidas neste trabalho lidam, especialmente, com problemas associados a fase de planejamento e controle operacional de SFM. Sendo assim, são abordados neste item as medidas de desempenho, os modelos e as ferramentas típicas para a avaliação e o tratamento de problemas afetos a esta fase do ciclo de vida destes sistemas.

2.4.1 Medidas de Desempenho

SURI [1985], abordando este assunto, discute as medidas de desempenho e os modelos de avaliação utilizados no tratamento dos problemas e decisões desta fase. Algumas das medidas de desempenho apontadas por SURI [1985] são apresentadas na tabela 2.2.

Medidas de Desempenho Típicas
Taxa de Utilização das Máquinas Volume de Produção Volume de Produtos em Processo Tempo de Fluxo de Peças Tempo do Ciclo de Produção Filas em cada Recurso Índices de Flexibilidade

Tab. 2.2: Medidas de Desempenho

2.4.2 Modelos para a Avaliação de SFM

A literatura contém vários exemplos da aplicação de modelos matemáticos, que foram apontados por SURI [1985], desenvolvidos para analisar, avaliar e solucionar problemas associados ao desempenho de SFM [BROWNE, 1984; BUZACOTT, 1986; CARRIE, 1985; MONTAZERI, 1990; SOLBERG, 1976; STECKE, 1981 e WHITNEY, 1985].

Do ponto de vista prático, pode-se diferencia-los entre modelos matemáticos de otimização (encontram um candidato ótimo dentre um conjunto), e descritivos ou avaliativos (descrevem e permitem avaliar o comportamento do conjunto de candidatos). Programação matemática é um exemplo típico de modelos de otimização, enquanto redes de filas, análise de perturbações, redes de Petri e simulação, são exemplos típicos de modelos matemáticos descritivos ou avaliativos. Em uma comparação da aplicação destes modelos a problemas em SFM alguns autores descrevem como segue, o uso destas ferramentas.

Programação Matemática: CHENG [1985] descreve que, essencialmente, os modelos de otimização são utilizados para a solução de determinados tipos de problemas que, embora complexos, exigem um nível de suposições simplificadoras, as quais na melhor das

hipóteses, os tornam aptos a tratar de sistemas em estado de equilíbrio.

Modelos de Redes de Filas: Estes modelos consideram a dinâmica, as interações e as incertezas no sistema, mas de forma agregada e não consideram certas características (detalhes) do mesmo. Além disso, os resultados são valores médios os quais assumem um sistema em seu estado de equilíbrio. Entretanto, os modelos tendem a apresentar estimativas razoáveis do desempenho do sistema, requerendo pouca quantidade de dados de entrada, bem como tempo de processamento computacional. Um modelo típico de redes de filas costuma requerer 20 à 40 itens de dados de entrada e tomam de 1 a 10 seg. de tempo de processamento em um microcomputador.

Modelos de Redes de Filas podem ser usados interativamente para chegar-se a resultados preliminares. Modelos então mais refinados, podem ser gerados para tomada de decisões.

O primeiro dos modelos de redes de filas para SFM, chamado CAN-Q, foi desenvolvido por SOLBERG [1980]. Mais recentemente, SURI e HILDEBRAND [SURI, 1984] descrevem o MVAQ, o qual pode modelar com mais detalhes que o CAN-Q sem perda de eficiência. Outra abordagem usando modelos de filas foi a realizada por ZHUANG[1993], já anteriormente citada.

Uma discussão detalhada da aplicação de modelos de Redes de Filas para SFM e sua comparação com simulação, também pode ser encontrada na mesma referência.

Modelos de Análise de Perturbações: Esta é uma técnica relativamente nova, com aplicações potenciais na simulação e em operações em SFM em tempo-real. A idéia básica aqui é observar o comportamento detalhado do sistema - via simulação ou sistema real - para um conjunto de parâmetros de decisão. Efetuando-se alguns cálculos adicionais, enquanto o sistema está sendo observado, os modelos de análise de perturbações podem prever o comportamento do sistema se estes parâmetros forem alterados. O importante é que não existe a necessidade de observar o sistema novamente (ou simular novamente), todas as predições são obtidas de uma única observação. Claramente isto pode evitar longos tempos de simulação ou ajudar o gerente de produção a melhorar suas decisões sem a necessidade de novos experimentos no próprio sistema.

A maior desvantagem destes modelos, fica por conta da falta de previsão mais acurada quando de grandes modificações nas variáveis de decisão. Sendo, no entanto extremamente válidos para procedimentos de "ajuste fino" em decisões relativas a projetos e ou procedimentos operacionais de SFM. Uma revisão dos modelos de análise de perturbações é dada por HO [1985]. Já suas aplicações em SFM podem ser vistos em Suri e Dille [SURI, 1985].

Redes de Petri: O atual estado da arte na aplicação das redes de Petri a problemas em SFM, permite uma modelagem determinística dinâmica dos sistemas. Enquanto no passado recente, esta teoria era usada para responder questões qualitativas, avanços recentes em redes de Petri temporais, permitem também respostas a questões de ordem quantitativa relacionadas ao desempenho do sistema. Existe também a possibilidade destas serem utilizadas para o controle de situações em tempo real. Entretanto, permanecem algumas questões relacionadas com a eficiência da aplicação desta técnica, quando se exige um maior nível de detalhamento do sistema, tais como várias máquinas com buffers finitos e políticas de roteiros em tempo real. Além disso, os atuais modelos não incorporam nenhuma incerteza. Uma visão geral de seu uso é encontrado em DUBOIS[1983]. Já sua aplicação à modelagem e análise de desempenho de SFM pode ser visto em YIM [1993].

Simulação: Como modelo avaliativo, a simulação tem sido a ferramenta mais utilizada na solução de problemas vinculados a projetos de SFM. Tais modelos podem imitar detalhadamente a operação destes sistemas por meio de um programa computacional. Os modelos de simulação podem ser muito ou pouco detalhados, dependendo do propósito de seu uso. O preço a ser pago por um nível muito grande de detalhes fica por conta da complexidade e tempo de modelagem e tempo de processamento. Os modelos de simulação são utilizados neste trabalho como meio de avaliar o desempenho operacional de SFM. Este motivo determina uma revisão da literatura sobre o tema o qual é apresentado a seguir.

2.5 Simulação de Sistemas Flexíveis de Manufatura

Como técnica de modelagem, a simulação computacional pode ser usada para analisar e ajudar a solucionar muitos problemas associados com as várias fases do ciclo de vida de um SFM.

Em um artigo de Martin e Musselman [MARTIN, 1984], foram identificadas quatro fases no ciclo de vida de todos os sistemas de manufatura. Estas fases em ordem de ocorrência são: projeto, justificação, implementação e operação. Os autores demonstram através do uso de quatro estudos de casos, que a simulação computacional é versátil e eficiente em todas as várias fases. Cada um dos quatro casos representam uma fase no ciclo de vida de um SFM.

No primeiro caso, um projeto inicial de um sistema que produz uma variedade de pistões é proposto. Um modelo de simulação é então construído para avaliar o impacto de modificações no projeto (por exemplo, a colocação de pequenas áreas de estoques entre máquinas e a divisão de uma operação com grande carga de peças em mais operações com cargas pequenas), sobre o tempo de processamento das peças ou sobre o total de investimento de capital.

O segundo estudo mostra, através da simulação, como uma proposta de modificação em um sistema existente pode ser justificada. Nesta comparação, uma determinada operação é substituída por dois processos automatizados. O desempenho do sistema é então avaliado com base em três critérios: inventário de peças em processo, tempo de entrega das peças e tempo total de processamento. Ao final do trabalho, a proposta de modificação é aceita com base nos resultados da simulação.

Um exemplo de como a simulação pode ser usada na fase de implementação de SFM é apresentado no terceiro estudo de caso, em que os autores mostram como a experimentação com um modelo de simulação, pode ser usado para determinar a necessidade de estrados e fixadores, variáveis consideradas críticas em SFM.

O último caso, fornece um exemplo de como se pode avaliar, via simulação, a capacidade do sistema com vários mix de produtos e sob diferentes condições operacionais (por exemplo no sequenciamento e carregamento de equipamentos).

Uma das vantagens da simulação sobre outros modelos matemáticos, é sua habilidade de modelar problemas dependentes do tempo, que, analiticamente tornam-se intratáveis. Entretanto, é conveniente lembrar que modelos de simulação podem tornar-se os mais dispendiosos, em relação a tempo e dinheiro se a linguagem e/ou a capacidade de modelagem apropriada não estiver disponível.

Em uma revisão sobre o uso da simulação como modelo matemático para analisar problemas em SFM, VIALI [1991], verificou que até meados da década passada, os SFM eram modelados usando-se linguagens de simulação de propósito geral, tais como GASP IV, SLAN II, SIMSCRIPT ou GPSS. Ultimamente, no entanto, muitas linguagens especialmente voltadas para a simulação de sistemas de manufatura tem sido desenvolvidas e usadas para àquele propósito. Dentre estas, pode-se citar MicroNet, SIMFACTORY II.5, SEE-WHY ou SIMAN IV.

O nível de detalhe a ser incorporado ao modelo, vai depender do nível de detalhe desejado na análise e, fundamentalmente, do seu propósito. CHENG [1985], identificou dois níveis de definição do sistema para avaliar alternativas de SFM usando-se simulação computacional. Num nível macro, um modelo mais grosseiro é suficiente para avaliar características mais gerais do sistema, tais como, utilizações de estações de trabalho, taxa de produção ou tamanho de filas. Neste caso, o analista está preocupado em avaliar diferentes alternativas do sistema em um estado de equilíbrio, para, por exemplo, determinar um projeto inicial viável que possa, posteriormente, ser examinado com mais detalhes. Já num nível micro, um modelo de simulação mais detalhado é desenvolvido para determinar, por exemplo, necessidades de equipamentos ou problemas de gargalos de produção. Da mesma forma, problemas operacionais, tais como, número de estrados ou fixadores, políticas de carregamento ou regras para sequenciamento, devem ser também investigadas neste nível, uma vez que um projeto apropriado tenha sido definido no nível macro.

2.5.1 Simulação Aplicada ao Desenvolvimento de Projetos de SFM

A literatura apresenta vários exemplos da utilização da simulação como ferramenta para o desenvolvimento de projetos de SFM.

Kiran e outros [KIRAM, 1989], relatam a aplicação de modelos de simulação num processo de escolha de um entre três diferentes projetos de SFM para a fabricação de 37 diferentes tipos de componentes agrupados em 8 famílias, para uma indústria aeronáutica. A simulação é usada na fase final do processo de decisão. Para a triagem inicial foram utilizados modelos de redes de filas. Os critérios para avaliar o desempenho das diferentes configurações foram: tempo médio das peças no sistema, número médio de peças nas filas e utilização das máquinas. Os resultados foram também significantes para a avaliação do desempenho e capacidade individual de certos componentes dos sistemas. Ao final do processo de simulação técnicas estatísticas foram utilizadas para a análise e validação dos resultados.

Em um dos trabalhos mais citados na literatura, Allan Carrie [CARRIE, 1986], discute o papel da simulação no auxílio ao tratamento de problemas nas fases de projeto, planejamento e operação de SFM. Em se tratando de projeto, ele descreve todas as etapas de seu desenvolvimento, desde as propostas preliminares até o detalhamento do projeto escolhido, deixando claro a utilidade da simulação como ferramenta de avaliação pela possibilidade incremental de seu uso. Em um exemplo baseado numa aplicação real, são avaliadas propostas iniciais de projetos de SFM, definido o melhor projeto e detalhado o seu planejamento e controle da produção, com modelos de simulação que vão se sofisticando na medida das necessidades.

Outro trabalho que consideramos importante foi o realizado por WAHAB [1986] O autor em sua pesquisa de doutoramento, apresentou uma metodologia para refinamento de projetos de SFM a qual, a partir de um projeto inicial, de uma lista de equipamentos substitutos e de algumas regras disponíveis em um sistemas especialista para a detecção de gargalos, realiza análises, melhorando, na medida do possível, o projeto inicial, pela substituição de equipamentos. A proposta não considera, no entanto, alternativas de diferentes estratégias de manufatura. O processo é iterativo e continua até que algumas metas de produção tenham sido alcançadas ou que os equipamentos, ainda disponíveis na

lista não possam mais melhorar o desempenho do sistema. Este trabalho foi importante, na medida que se constitui uma das primeiras tentativas de empregar simulação e sistemas especialistas, em conjunto, para a solução de problemas em projetos de SFM.

2.5.2 Simulação Aplicada ao Planejamento e Controle da Produção de SFM

Muitas outras aplicações de simulação de SFM podem ser encontradas na literatura, especialmente àquelas voltadas para a solução de problemas afetos ao planejamento e ao controle de suas operações.

Um exemplo deste tipo de aplicação pode ser vista em CARLIER [1988]. Neste trabalho o autor apresenta a utilização de um modelo de simulação, para testar várias estratégias associadas ao planejamento da produção de um SFM real, objetivando a fabricação de peças para seis diferentes tipos de compressores. A grande variedade de peças, tempos de processamento e rotas de processamento, além de restrições no número de diferentes estrados e fixadores e, o compartilhamento de ferramentas, torna extremamente complexa a tarefa de determinar a sequência de entrada das peças no sistema e a designação de prioridades. Com esta ferramenta de avaliação disponível, os operadores podem testar várias alternativas de cenários, antes de iniciar um ciclo de produção, o qual pode durar de algumas horas a alguns dias.

Outro exemplo do uso de modelos de simulação com este propósito é o trabalho de WANG [1986] onde a autora analisa o desempenho de um SFM com três diferentes configurações. O objetivo é verificar o impacto de várias alternativas no planejamento das operações considerando, o número de pallets, o grupamento de máquinas, o número de veículos transportadores e variações na demanda. Após simular um grande número de experimentos com várias combinações entre as variáveis e realizar análises estatísticas (análise de variância para médias e regressões múltiplas) Wang conclui que o grupamento de máquinas possui um grande impacto sobre o desempenho do sistema. A combinação número de transportes/pallets/política de prioridades também exerce grande influência. A variação na demanda parece, segundo suas conclusões, não alterar significativamente o desempenho do sistema, de acordo com os critérios adotados.

São comuns também a utilização da simulação em estudos de avaliação de diferentes políticas operacionais a SFM. Mishra e outros. [MISHRA, 1988], avaliam o desempenho do sistema através dos seguintes critérios (medidas de desempenho): número de peças em atraso, número de peças completadas, número de peças em processamento e taxa de utilização das máquinas. Os resultados das simulações efetuadas, demonstram a utilidade de algumas das políticas de programação aplicadas bem como a necessidade de um sistema de transporte mais rápido. Foi testado também a influência da quebra de equipamentos no desempenho do sistema.

Importante citar também o trabalho de MAHESHWARI [1992], que usando modelos de simulação, testa um conjunto de ações integradas para o planejamento e controle da produção de SFM, contribuindo para uma melhor compreensão das influências exercidas por algumas variáveis (número e tipos de pallets, tamanho dos buffers nos centros de usinagem, sequenciamento, prioridades, localização de ferramentas, etc) no desempenho operacional destes sistemas.

Em um trabalho bastante abrangente, Gupta e Goyal [GUPTA, 1992] apresentaram um estudo empírico sobre características físicas [composição e layout] de SFM, combinadas com diversas regras de gerenciamento da produção (planejamento e operação). O objetivo do trabalho é a avaliação do impacto dos diversos tipos e graus de flexibilidades associadas aos SFM, sobre sua performance. Os autores, concordando com Hayes e Wheelwright [HAYES, 1984], acham que os SFM oferecem grande vantagem competitiva às empresas, operando com diversificação e baixo volume sobre produtos com pequeno ciclo de vida. No entanto, ressaltam que o nível de flexibilidade imposto ao sistema deve ser gerenciado desde o projeto até a operação destes sistemas, uma vez que muitas das flexibilidades são consideradas antagônicas, tornando imperativo à gerência reconhecer quais são críticas à estratégia de manufatura da empresa. Em sua pesquisa várias configurações de SFM foram simuladas sobre a influência de uma grande combinação de estratégias de carregamento e despacho. Os resultados mostraram que é possível encontrar um meio-termo entre os vários tipos de flexibilidades sobre várias circunstâncias. Alguns cenários ilustram que uma combinação apropriada de variáveis, tais como, programação, variedade de produtos e configuração do sistema podem ser muito mais efetivas que outras ao conter as flexibilidades impostas. Foi mostrado também que os SFM podem conter com êxito medido, vários tipos de flexibilidade simultaneamente.

2.5.3 Avanços no Uso da Simulação na Modelagem de SFM

Muitos esforços têm sido feitos para desenvolver modelos de simulação de propósito geral capazes de auxiliar no projeto e análise de SFM. O produto final de tais esforços é, usualmente, um pacote computacional amigável, que permite ao usuário, com pouco conhecimento de simulação computacional, especificar o sistema sob investigação e obter as medidas de desempenho pertinentes com relativa facilidade.

Elmaraghi e Ho [ELMARAGHY, 1982] desenvolveram um simulador de SFM - FMS-SIM, que orientado pelo usuário pode ser usado para testar projetos, estudar efeitos de estratégias de controle, regras e prioridades no sequenciamento e/ou na programação, além de arranjo físico. Este pacote executa a simulação, via construção de um modelo apropriado, a partir de uma biblioteca de subrotinas especialmente construídas. Os resultados da simulação podem ser apresentados na forma de relatórios convencionais ou animação gráfica pós-processamento, isto é apresenta o movimento das peças sendo processadas, após o término da rodada de simulação. Este tipo de saída é muito interessante pois auxilia tanto na validação do modelo, isto é, testar se o mesmo opera de acordo com o que foi proposto, como na avaliação visual do desempenho do sistema que se está simulando. Um exemplo típico de seu uso está na detecção de gargalos na produção ou como forma de estímulo na justificativa de novos projetos.

Devido ao pouco desenvolvimento de sistemas de computação gráfica naquela época, a animação desenvolvida era ainda muito rudimentar.

Medeiros e Sadowski [MEDEIROS, 1983] apresentaram um pacote de modelagem para análise de projetos de células de manufatura controladas por computador. A abordagem é semelhante à feita por Elmaraghi e Ho. Neste caso, o sistema baseia-se na linguagem de simulação Q-GERT e apresenta opções entre rotinas já disponíveis ou aquelas desenvolvidas pelo próprio usuário. O pacote é limitado, uma vez que a linguagem na qual foi baseada é mais voltada para teoria das filas do que simulação e portanto, não permite níveis mais profundos de detalhamento.

Com a evolução dos equipamentos de informática, especialmente dos micro computadores, muitos projetos desenvolvidos para equipamentos de grande porte, possuem agora versões adaptadas a equipamentos pessoais, tornando-se grande seu uso e divulgação. Revisões mais recentes, de softwares disponíveis para simulação de sistemas de manufatura podem ser vistas em CARRIE [1988]; GREENWOOD [1988] e VIALI [1991]. Nomes como SIMAN/CINEMA, AUTOGRAN, PCMODEL, SEE-WHY, SYMFACTORY e outros, são motivos de palestras, cursos, congressos e publicações, tratando de suas virtudes, possibilidades e aplicações. Na sua grande maioria possuem versões poderosas também para computadores pessoais e se utilizam de interfaces gráficas na comunicação com os usuários nas fases de modelagem, execução e apresentação de resultados.

Além dos avanços das linguagens de simulação voltadas especialmente a simulação de sistemas de manufatura, incluindo-se os SFM, observa-se alguns esforços para a integração com outras áreas do conhecimento no sentido de facilitar o uso de modelos de simulação. Dentre estas tecnologias, cita-se, principalmente, os sistemas especialistas. A literatura sobre sistemas de apoio a decisão, envolvendo estas duas áreas, os chamados sistemas inteligentes de simulação, é tratada no último tópico desta revisão.

2.6 Sistemas Especialistas

2.6.1 Definição

Harmon e King [HARMON, 1988] apresentam a seguinte definição de um sistema especialista:

“...é um programa inteligente de computação, que usa conhecimentos e procedimentos de inferência, para resolver problemas que requerem o conhecimento e a experiência humana na sua solução. O conhecimento necessário mais os procedimentos de inferência fazem-no atuar como os melhores praticantes no campo de conhecimento associado”.

O que Harmon e King apresentam como “conhecimento” pode ser definido como um conjunto de fatos e heurísticas incorporados ao sistema. Os fatos são constituídos de um

corpo de informações que pode ser amplamente compartilhado, isto é, de domínio público e, geralmente aceito por especialistas do ramo. Com relação as heurísticas, estas são geralmente privadas, isto é, constituem-se de regras de raciocínio plausível ou de bom senso pouco discutidas. Estas regras caracterizam o nível de especialista nas tomadas de decisão. Segundo os autores, o nível de desempenho de um sistema especialista depende do tamanho e da qualidade da base de conhecimentos.

A pesquisa e o desenvolvimento de sistemas especialistas é um dos principais ramos da inteligência artificial. Outros são as áreas de processamento da linguagem natural, robótica e sistemas visuais, redes neurais. etc.. As pesquisas em inteligência artificial preocupam-se, principalmente, com idéias e modos de tornar os computadores inteligentes [WINSTON, 1984]. Já as pesquisas em sistemas especialistas, concentram-se no desenvolvimento de programas computacionais de alto desempenho, em campos profissionais especializados, que imitam o comportamento de especialistas humanos em seus procedimentos na solução de problemas específicos [HAYES, 1983].

Alguns sistemas especialistas possuem desempenho igual ou melhor do que os próprios especialistas, em algumas áreas de conhecimento. Outros, no entanto não alcançaram este nível de desempenho. Segundo WATERMAN [1986], apesar dos sistemas especialistas apresentarem, cada vez mais, melhores desempenhos, em algumas áreas a mente humana é claramente superior. No entanto, segue afirmando, isto não reflete uma deficiência ou limitação da inteligência artificial, mas apenas um estado da arte. Atualmente as pesquisas na direção das redes neurais, parecem confirmar as palavras do Prof. Waterman.

2.6.2 Diferenças entre Sistemas Especialistas e a Programação Convencional

As aplicações da inteligência artificial envolvem muitos fatores que a distingue da programação convencional, tais como a representação simbólica e a pesquisa heurística. Os sistemas especialistas, como uma área da inteligência artificial, apresentam diferenças adicionais. HARMON [1988], SCOTT [1991] e CHORAFAS [1990] citam algumas diferenças das quais apresentamos uma compilação:

1. Os sistemas especialistas desempenham tarefas antes desempenhadas por especialistas humanos:

2. Os sistemas especialistas são mantidos por especialistas do campo de conhecimento associado e por engenheiros de conhecimento, enquanto que os programas convencionais são mantidos por programadores e analistas;
3. A base de conhecimentos de um sistema especialista encontra-se em uma forma compreensível e pode ser modificada facilmente;
4. A programação convencional usa algoritmos em sua estrutura, enquanto os sistemas especialistas fazem uso de heurísticas e máquinas de inferência.

Os novos conceitos envolvidos na pesquisa de sistemas especialistas, enfatizam os aspectos cognitivos do pensamento humano relacionados ao ato de decidir, isto é, o foco principal dos sistemas é imitar a experiência e o conhecimento do decisor humano.

2.6.3 Estrutura de um Sistema Especialista

Nesta seção são revistos alguns textos que descrevem a estrutura de um S.E., isto é: a base de conhecimentos e a máquina de inferência. A base de conhecimentos contém a representação do domínio do conhecimento que, no protótipo está associado à análise e a avaliação de SFM. Este conhecimento é transcrito para o sistema pelo uso de duas técnicas de representação: frames e regras de produção. A máquina de inferência, por sua vez, contém o conhecimento geral que direciona e controla o raciocínio do sistema.

2.6.3.1 A Base de Conhecimentos

A discussão que segue sobre a base de conhecimentos, estará focando os dois métodos citados para a representação do conhecimento.

1 Representação do conhecimento sob a forma de Regras de Produção:

A representação de conhecimentos sob a forma de regras de produção é, no momento a forma mais utilizada dentro de S.E.. Em geral uma regra de produção é representada por um conjunto de condições, conclusões e ações. Se as condições são verdadeiras, a regra é disparada e as ações são desencadeadas [HARMON, 1988; SCOTT, 1991].

Regras de produção são, em geral, facilmente entendidas e implementadas por especialistas no domínio do conhecimento e possuem um poder de expressão suficiente para representarem este domínio. Por si próprias, no entanto, as regras de produção não são um meio efetivo de representação para a maioria dos sistemas baseados em conhecimentos. Em particular, seu poder de expressão é inadequado para a definição de termos e para descrição de objetos e suas relações. As áreas onde as regras de produção são mais inadequadas, são justamente àquelas melhor cobertas pelos frames [HAYES, 1983 e WALTERMAN, 1986].

2 *Representação do conhecimento sob forma de Frames:*

O conceito de *frame* foi primeiramente apresentado por Marvin Minsky, conforme citado por Fikes e Kehler [FIKES, 1985]. Os frames são estruturas usadas para representar objetos ou conceitos, através de seu conjunto de atributos e métodos. Genericamente, um frame é constituído de um nome, um conjunto de *slots* (atributos), com os respectivos valores que caracterizam o conceito e, finalmente, um conjunto de métodos para a utilização do conceito representado e para a sua atuação e comunicação. A representação de conhecimentos através de frames, tal como na programação orientada à objetos (POO), assenta-se na noção de conceitos e hierarquia de conceitos.

Os frames proporcionam uma rica estrutura na qual se podem definir objetos ou mais especificamente seu atributos que mais tarde serão tratados dentro das regras de produção. Estas estruturas podem também serem usadas para um particionamento ou para a formação de um índice das regras de produção do sistema. Esta capacidade facilita ao especialista, o entendimento e a construção de regras e ao projetista controlar quando e com que propósito uma coleção particular de regras são usadas pelo sistema.

Algumas vezes, a semelhança da POO, os frames são referidos como classes. Assim, objetos são definidos por meio de classes e instâncias, nos quais um conjunto de atributos (*slots*) e métodos associados à estes atributos estruturam toda a informação relativa aos objetos definidos, bem como toda a comunicação destes com outros objetos do sistema, por meio de mensagens.

produção, como o fazem, no nível operacional. Fazendo-se uma comparação, os métodos desempenham um papel parecido com o de subrotinas em programas convencionais e podem alocar os resultados de suas avaliações aos atributos de outras classes.

2.6.3.2 A Máquina de Inferência

A máquina de inferência contém o conhecimento geral que direciona e controla o raciocínio do sistema. O mecanismo de inferência, executa duas tarefas da maior importância. Primeiro examina os fatos e regras existentes na base de dados, acrescentando novos fatos sempre que possível, na medida das interpretações e conclusões que executa. Segundo, decide a ordem com que tais inferências são tiradas, por meio de seu mecanismo de controle [FISCHLER, 1987, HARMON, 1988, INTELLICORP, 1992].

Os sistemas especialistas usam em seus mecanismos de inferência dois tipos de estratégias de controle: o *encadeamento ou busca para frente* e o *encadeamento ou busca para trás*. Ambos podem ser usados sobre a mesma base de conhecimentos e a opção por uma ou outra estratégia, dependerá, basicamente, da existência desta possibilidade na ferramenta em que o sistema foi desenvolvido e, principalmente, do tipo de objetivo buscado.

Encadeamento para Frente

Este procedimento, em geral, procura descobrir as conseqüências de uma nova informação ou dado pela comparação da parte SE. de uma regra ao fato novo. Se o este fato coincidir (casar) com o padrão de uma das premissas da parte SE e se todas as outras expressões da parte SE desta regra forem verificadas, então a regra pode ser aplicada. Isto significa que as ações representadas por expressões na parte ENTÃO da regra são executadas. Normalmente, estas ações adicionam novas informações ao sistema. Toda vez que uma regra é aplicada durante um *encadeamento ou busca para frente*, o fato ou informação altera o valor de um ou mais atributos ou slots do sistema. Este novo valor, por sua vez, torna-se um fato novo iniciando um processo de busca mais profundo. Este ciclo repete-se até que certas condições sejam encontradas ou que mais nenhuma regra possa ser ativada.

parte SE desta regra forem verificadas, então a regra pode ser aplicada. Isto significa que as ações representadas por expressões na parte ENTÃO da regra são executadas. Normalmente, estas ações adicionam novas informações ao sistema. Toda vez que uma regra é aplicada durante um *encadeamento ou busca para frente*, o fato ou informação altera o valor de um ou mais atributos ou slots do sistema. Este novo valor, por sua vez, torna-se um fato novo iniciando um processo de busca mais profundo. Este ciclo repete-se até que certas condições sejam encontradas ou que mais nenhuma regra possa ser ativada.

No *encadeamento ou busca para frente*, quatro estruturas de dados desempenham um importante papel:

Agenda: - A agenda é representada por uma fila de pares objeto:atributo, processados pela máquina de inferência. Sempre que o valor de um atributo é alterado (fato novo) pela aplicação de uma regra, este é colocado automaticamente na agenda. Uma forma de interrupção do processo de busca é não existir nenhum par de objeto:atributo na agenda ou no conjunto de regras.

Conjunto de Regras: - É formado pelo grupo de regras que podem ser usadas no *encadeamento para frente*. Usualmente, todas as regras fazem parte deste conjunto, no entanto, isto pode ser restringido. De acordo com os objetivos da busca, pode-se permitir que apenas uma parte do universo de regras sejam utilizadas.

Lista de Regras Ativas - É a lista de todas as regras, pertencentes ao *Conjunto de Regras*, que possuem uma premissa coincidindo com um item da agenda. A prioridade no exame das regras desta lista é dada, tanto pelas prioridades individuais de cada uma delas, como por uma estratégia geral resolução de conflitos que pode ser determinada para definir a forma de ação da máquina de inferência.

Objetivo: - O objetivo, que pode ser opcional, é em geral, formado por uma expressão, uma espécie de teste de parada. Sempre que este for verificado, a busca é interrompida, não importando o estado da agenda.

A *busca para frente* é um processo cíclico com os seguintes passos básicos:

1. O primeiro par objeto:atributo, da *Agenda* é comparado com todas as regras do *Conjunto de Regras* verificando-se coincidências (casamentos) nas premissas;
2. Qualquer regra coincidente é ordenada, de acordo com as prioridades e adicionada à *Lista de Regras Ativas*;
3. Testa-se a primeira regra da *Lista de Regras Ativas* ou avalia-se o primeiro item da *Agenda* (como no passo 1), dependendo do tipo de estratégia de avaliação adotado e do estado de cada fila:
 - a) Se a *Agenda* está vazia, testa-se a primeira regra na *Lista de Regras Ativas*;
 - b) Se a *Lista de Regras Ativas* está vazia, avalia-se o primeiro item da *Agenda*;
 - c) Se existirem itens em ambas as filas, utilize a estratégia de avaliação adotada para decidir;
- 4) Se todas as expressões na parte SE da regra são verdadeiras, executa-se as ações especificadas na parte ENTÃO da regra.

Encadeamento para Trás

No *encadeamento ou busca para trás*, também conhecido como *raciocínio dirigido por objetivo*, a máquina de inferência procura alcançar um determinado *Objetivo*, buscando regras que possam provar o fato e depois tentando provar suas premissas. Estas premissas, por seu turno, tornam-se novos fatos a serem verificados por novas regras e assim por diante. Nesta forma de *raciocínio*, o mecanismo de inferência tenta verificar uma hipótese pela comparação a parte ENTÃO de uma regra com o *Objetivo* que, neste caso, funciona como um disparador de todo o processo.

Se o *Objetivo* casa com uma das expressões na parte ENTÃO de uma regra e se todas as expressões na parte SE da regra são verificáveis, então esta pode ser aplicada. Toda vez que uma premissa de uma regra não puder ser imediatamente verificada, a própria premissa passa a ser um novo *Objetivo* a ser resolvido. Este ciclo é repetindo até que todos os objetivos sejam resolvidos, criando-se neste processo uma cadeia cujo início é um objetivo e cujo comprimento depende do número de fatos adicionais inferidos pelas regras. No *encadeamento ou busca para trás*, tem-se também algumas estruturas de dados jogando um importante papel:

- *Conjunto de Regras e Lista de Regras Ativas*- Semelhantes ao usado no encadeamento para frente.
- *Objetivo* - Uma expressão representando uma hipótese a ser verificada. Sempre que este puder ser provado, o processo é considerado um sucesso.
- *Pilha de Objetivos* - Um conjunto de pares Objeto:Atributo a serem verificados. Atributos cujos valores não podem ser imediatamente verificados olhando-se a base de conhecimentos, são adicionados à pilha como novos objetivos.

O processo de *Encadeamento ou busca para trás*, inicia questionando a existência de um objetivo. Não existindo um, o processo termina sem iniciar uma busca. Existindo um objetivo, o mesmo é avaliado. Se todos os atributos necessários à avaliação são conhecidos, o processo retorna um valor, VERDADE OU FALSO.

A existência de atributos desconhecidos para a avaliação, torna os novos objetos passíveis de avaliação, sendo estes colocados na pilha de objetivos. Após este passo inicial, inicia-se o Ciclo de Expansão. Neste ciclo, são verificadas as conclusões de todas as regras permitidas pelo *Conjunto de Regras*, que casem com os pares objeto:atributo da pilha de objetivos. Para cada regra verificada, os pares objeto:atributo de suas premissas, são verificados e todos os desconhecidos são também colocados na pilha de objetivos. Ao final deste ciclo, tem-se um conjunto de regras ativas e um conjunto de objetivos para serem testados. Durante este processo de expansão, novas informações são constantemente coletadas e muitos pares objeto:atributo, antes sem valor, podem agora ter respostas. Se, após este processo de varredura sobre todas as regras da *Lista de Regras Ativas*, em que novas informações são coletadas, permitindo a retirada de pares objeto:atributo da pilha de objetivos, ainda restarem pares sem possibilidade de avaliação, o processo prossegue para o Ciclo de Inquisição. Neste ciclo, todos os pares objeto:atributo desconhecidos restantes na pilha de objetivos são processados por meio de um questionamento ao usuário. Nesta última tentativa de concluir a busca, o usuário poderá responder trazendo fatos novos a base de conhecimentos ou simplesmente responder que desconhece os valores perguntados. Finalmente, com base nas respostas do usuário, será possível concluir a busca. O retorno será uma avaliação do objetivo inicial, cuja resposta será VERDADE ou FALSO.

2.6.4 Ferramentas para o Desenvolvimento de Sistemas Especialistas

Um levantamento feito por Harmon e King [HARMON, 1988] e Waterman [1986], aponta, de maneira geral, para uma divisão em três categorias distintas, das ferramentas disponíveis para o desenvolvimento de sistemas especialistas:

1. Ferramentas para a construção de pequenos sistemas, com até 50 regras, as quais podem ser usadas em computadores pessoais: Expert-Ease e ES/P ADVISOR, são exemplos típicos;
2. Ferramentas para a construção de sistemas restritos, de médio e grande porte, com mais de 500 regras. Estes sistemas requerem equipamentos tipo super-micros ou estações de trabalho e costumam estar restritos a consultas de um tipo de paradigma, tais como planejamento, projeto ou diagnóstico. EXPERT, KESS e OPS 5, são exemplos desta categoria de ferramentas para o desenvolvimento destes sistemas.
3. Ferramentas para o desenvolvimento de grandes sistemas híbridos. Os requerimentos básicos, são os mesmos da categoria anterior. A diferença fica por conta das características de consultas aqui permitidas, não mais restritas a um único paradigma. Exemplos de ferramentas disponíveis nesta categoria são o ART e o KEE.

Além destas ferramentas, existem algumas linguagens especialmente voltadas para a construção de sistemas especialistas, tais como o LISP e o PROLOG. Muitas dessas ferramentas geram seus códigos numa destas linguagens. Um exemplo é o KEE (Knowledge Engineering Environment) [INTELLICORP, 1985], que gera um código em LISP. A definição sobre o que utilizar para o desenvolvimento de um sistema especialista segue basicamente as regras gerais da programação de computadores, isto é, a facilidade de adaptação do usuário e a inter-relação, entre a ferramenta ou a linguagem e o propósito do sistema a ser gerado.

Um grande sucesso tem sido alcançado, também, por ferramentas que permitem integrar frames e regras de produção para formar sistemas híbridos, que combinam as vantagens

das duas técnicas de representação do conhecimento. Seu maior representante é o KEE voltado para sistemas de grande porte implantados especialmente em estações de trabalho e mais recentemente o Kappa-PC [INTELLICORP, 1992], voltado ao segmento dos computadores pessoais.

A novidade destes ambientes de desenvolvimento de sistemas, principalmente daqueles voltados ao segmento dos computadores pessoais, é sua extrema facilidade de uso, versatilidade e portabilidade, devido principalmente ao ambiente operacional ao qual está integrado (Windows). O KAPPA, faz uso de muitas facilidades gráficas herdadas do KEE. Na verdade o KAPPA é descrito como uma ferramenta gráfica orientada a objetos para o desenvolvimento de sistemas especialistas.

Pela suas características e facilidades, o KAPPA, já agora na sua versão 2.0, foi a software básico escolhido para o desenvolvimento, tanto do sistema especialista quanto da interface gráfica com o usuário, embutidos na ferramenta proposta neste trabalho.

2.7 Sistemas Inteligentes de Simulação (SIS)

A integração de sistemas especialistas com técnicas de simulação tem permitido a aceitação e o uso dos SIS como uma potente ferramenta para o planejamento, o projeto e o controle de complexos sistemas de produção [FREITAS, 1990].

De fato, nos últimos anos, a simulação deixou de ser uma ferramenta tida como "último recurso", para tornar-se um instrumento de inestimável valor no auxílio à solução de problemas relacionados com a engenharia, projeto e gerenciamento da produção.

O fato de ter sido, até recentemente, rechaçada como recurso auxiliar à tomada de decisões, deveu-se, principalmente, a necessidade de:

- a) recursos humanos especializados;
- b) razoável suporte computacional e
- c) uma grande disponibilidade de tempo, principalmente na fase de modelagem.

Desta forma, o emprego de novos recursos na pesquisa e desenvolvimento das linguagens de simulação tem levado ao rápido crescimento, tanto de suas capacidades, quanto de seus usos.

Segundo SHANNOM [1988], os objetivos destas pesquisas são três:

- a) redução do tempo de modelagem;
- b) melhoramento da precisão dos resultados e
- c) melhoramento da comunicação usuário/sistema.

Tais objetivos têm sido parcialmente obtidos, devido à criatividade dos pesquisadores no desenvolvimento de linguagens e pacotes de simulação, que fazem uso de recursos gráficos e de sistemas especialistas, e de razoáveis recursos computacionais.

2.7.1 As Similaridades entre Sistemas Especialistas e Simulação

As técnicas de simulação e os sistemas especialistas apresentam certas similaridades. Gaines e Shaw [GAINES, 1985], por exemplo sugerem que os S.E. atuam como se fossem simuladores de pessoas. Já O'Keefe [1986], aponta para as similaridades técnicas (representação modular e mecanismos de inferência) e funcional (função auxiliar aos processos decisórios).

Com base nestas similaridades entre simulação e sistemas especialistas, O'Keefe [1986], desenvolve e discute as maneiras pelas quais podem ser combinadas as duas técnicas. Ele apoia esta descrição em muitos exemplos. Suas sugestões são as seguintes:

1. *Embutido*: O sistema especialista está embutido no sistema de simulação ou vice-versa. Muitos sistemas de simulação usam conhecimentos ao invés de dados. Por exemplo, as regras de prioridades em uma fila podem ser consideradas conhecimentos, sendo talvez mais interessante embuti-las em uma base de conhecimentos do que em um código de simulação. Por outro

lado, um sistema especialista pode necessitar de uma simulação para atualizar uma variável dependente do tempo (elementos de difícil tratamento ao nível de sistemas especialistas).

2. *Paralelo:* Um modelo de simulação e um sistema especialista que tenham sido projetados, desenvolvidos e implementados separadamente, podem interagir, um interrogando o outro. Segundo O'Keefe, esta é uma área de crescente interesse entre os engenheiros de conhecimento, tendo em vista a disponibilidade de software em ambos os segmentos.
3. *Cooperativo:* Neste modo, o sistema especialista e o modelo de simulação podem, ambos, fazer parte de um grande sistema de apoio à decisão.
4. *Entrada e Saída Inteligente:* Neste caso, o sistema especialista conduz um diálogo com o usuário, de modo a gerar um código ou conjunto de instruções para serem usados por um pacote ou linguagem de simulação, diminuindo com isso o tempo de modelagem, a obtenção de resultados relevantes ou permitindo uma melhor interpretação e compreensão dos resultados.

Segundo outros autores [FREITAS, 1990; SABUNCUOGLU, 1988 e SHANNON, 1988], esta forma de interação é atualmente uma área de muita dedicação por parte de alguns centros de pesquisas.

Partindo das formas sugeridas por O'Keefe, para a combinação entre sistemas especialistas e sistemas de simulação, SHANNON [1985 e 1988], SABUNCUOGLU [1988] e o próprio O'KEEFE [1986], propõem modelos ideais de SIS, os quais se combinados, resultariam em um sistema com as seguintes características:

- a) habilidade de construir modelos de simulação sobre estudos e esforços de modelos passados, isto é, o sistema preservaria conhecimentos, aprendendo a cada estudo realizado;
- b) uma base de conhecimentos, consistindo de uma biblioteca de modelos com diferentes níveis de abstração;

- c) independência entre a estrutura geral de modelos e o detalhamento de dados necessários para descrever um modelo específico, isto é, inexisteriam formas padronizadas, o próprio sistema se encarregaria de encontrar o modelo;
- d) programação declarativa ao invés de processual;
- e) capacidade de autodirecionamento na busca de um objetivo especificado pelo usuário;
- f) capacidade de explicar as razões do não alcance de uma meta e/ou porque determinada solução é a mais recomendada;
- g) uma interface amigável de interação com o usuário, usando recursos gráficos, animação, diálogos em linguagem natural, etc..

Embora muitas das características citadas sejam passíveis de implementação, não conhecemos, ainda, uma ferramenta disponível no mercado, com tais capacidades.

Conforme citado por SHANNON [1988], os estudos em simulação, são dirigidos por metas ou objetivos, isto é, ninguém constrói um modelo pelo simples motivo de construí-lo. Tais objetivos ou metas, provém da necessidade dos decisores em analisar e entender problemas e/ou avaliar resultados decorrentes de determinadas intervenções. Um problema é criado pela percepção da necessidade de levar um sistema a um estado mais desejável. Uma intervenção pode ser uma mudança física, modificações nas ações de controle ou novas políticas e procedimentos.

Sendo portanto, os objetivos e metas que determinam o modelo, o tipo de experimento e a análise a ser feita, são eles que especificam que elementos do sistema real devem ser envolvidos, o propósito da análise, as variáveis de interesse e o grau de precisão dos resultados necessários.

Idealmente, o sistema deverá interpretar e entender as solicitações do usuário, por meio de linguagem natural ou outro meio, determinando então suas necessidades em termos de dados de entrada, técnicas de processamento da informação e formas de apresentação de resultados.

Neste sentido, existem algumas pesquisas no uso da linguagem natural, mas que, em parte, devido a suas atuais limitações, têm levado à popularização de interfaces gráficas

para a "tradução" das entradas do sistema desejadas pelo usuário. Desta forma escolhendo e colocando ícones pré-definidos e respondendo a algumas questões colocadas pelos programas, o usuário pode definir rapidamente um modelo de sistema para ser executado. Alguns destes programas, voltados para a simulação da manufatura, tais como o SIMFACTORY e o SEE-WHY, tem demonstrado que já é possível construir sistemas razoavelmente complexos sem escrever sequer uma linha de programação.

O impecílio ainda existente é a falta de flexibilidade nestes pacotes. Por exemplo no SIMFACTORY, não se pode criar um modelo de SFM, em que o sequenciamento no plano de produção de um tipo de peça não esteja previamente estabelecido. Desta forma, todas as peças seguem um determinado padrão, o qual não pode ser alterado de acordo, por exemplo com um determinado estado do sistema.

Algumas pesquisas tem procurado reunir as facilidades da modelagem visual com sistemas de IA. Este é o caso do trabalho de HURRION [1991]. Neste trabalho, o autor propõe um modelo de simulação visual interativa, o qual, com o auxílio de um módulo inteligente, isto é um sistema especialista, procura conduzir ou auxiliar o modelador nas tarefas de construção e/ou modificação de modelos. Nenhuma linguagem específica de simulação foi utilizada. O modelo de simulação foi desenvolvido em FORTRAN e o SE em PROLOG. Cada um dos programas roda em um microcomputador separado e são interligados por uma interface serial. Cada decisão necessária à simulação, é transferida ao sistema especialista no outro micro o qual devolve a solução por meio de um protocolo de transferencia de dados, na forma de texto e valores numéricos. Desta forma, por meio de trocas de parâmetros, o sistema especialista orienta o usuário na condução das modificações no modelo e nos experimentos. A separação dos dois modelos foi feita objetivando facilitar as alterações ou inclusões de regras em função dos experimentos.

2.7.2 O Uso de Sistemas Inteligentes de Simulação em SFM

Como qualquer outro sistema de produção, os SFM, também apresentam problemas relacionados a projeto, planejamento, programação ou controle, que a um nível mais geral, podem ser classificados como de cunho estratégico, tático ou operacional.

Muitos deles não são novos e aparecem também, freqüentemente na manufatura convencional. A novidade neste caso, relaciona-se aos riscos envolvidos em SFM tendo em vista a quantidade de capital envolvido e o longo tempo de implementação. Além disso, como sugere SUBUNCUOGLU [1988]:

"...qualquer problema com SFM é amplificado, tendo em vista suas necessidades básicas de grande flexibilidade e de sua natureza dinâmica."

Num contexto como este, os SIS aparecem como potentes ferramentas para lidar com inúmeros problemas afetos a complexos sistemas de manufatura.

De fato, os SFM, em particular e a manufatura integrada por computador (CIM), em geral, têm muito a se beneficiar com as possibilidades atuais e futuras dos SIS. Tal uso, permitirá ao analista dedicar-se mais ao problema ao invés de preocupar-se e diluir suas energias com as complexidades do modelo que o está auxiliando.

Muito embora tais perspectivas, o estado atual das tecnologias da inteligência artificial, em especial dos sistemas especialistas, aplicados à simulação ainda é incipiente e somente dois tipos de abordagens tem sido notadas. Na primeira, encontram-se os sistemas híbridos, que adotam entradas e saídas inteligentes sobre uma linguagem de simulação já existente. Já no segundo caso, encontram-se as pesquisas sobre novos sistemas, as quais adotam mudanças no paradigma da modelagem em simulação.

Com relação a primeira abordagem a idéia é desenvolver interfaces inteligentes de comunicação com o usuário, que permite a programação automática do sistema numa linguagem de simulação existente.

Tais sistemas, geralmente possuem um domínio fixo de atuação, tais como redes de computadores, montagens de circuitos eletrônicos, etc..

Alguma atenção tem sido dada também ao desenvolvimento de saídas inteligentes para os atuais sistemas e linguagens de simulação, auxiliando o usuário na análise dos resultados e sugerindo modificações. Nestes sistemas um objetivo é traçado e o modelo executado.

Se o resultado desejado não for alcançado, os sintomas relativos ao problema ocorrido são percebidos pelo sistema, que dispara, então uma série de sugestões ao usuário no sentido de corrigir o modelo, para que as distorções possam ser minimizadas. Usualmente, estas rotinas auxiliares são constituídas de uma série de regras do tipo *SE* este sintoma ou condição *ENTÃO* sugiro esta ação ou procedimento.

A vantagem dos sistemas híbridos é que eles não são tão difíceis de serem desenvolvidos e o modelo final apresenta, se bem desenvolvido, um bom desempenho computacional.

Com relação à segunda forma de abordagem, algumas tentativas bem sucedidas, levadas a efeito em renomados centros de pesquisa, tem ocorrido nos últimos dez anos, a partir do surgimento na década de 70 das linguagens de programação não procedurais. Um exemplo é o ROSS (Rule Oriented Simulation System) [KLAHR, 1980], desenvolvido pela Rand Corporation. O ROSS é um sistema interativo de simulação desenvolvido em LISP, considerando idéias que envolvem programação orientada a objeto (POO). Seu propósito é a simulação de jogos de guerra. Foi considerado um pioneiro no tratamento da realidade na forma de objetos.

Nesta forma de modelagem, a unidade de informação e de organização do sistema passam a ser objetos. O conjunto destas informações pode incluir fatos a respeito dos objetos e de como ele deve se comportar diante de certos estímulos e/ou com quem deve interagir. SHANNON [1985], afirma que o uso da POO nos SIS, é que permitirá levar um modelo e um experimento a um mais alto nível de abstração e a uma forma mais natural de representação. Segundo ele, os sistemas do mundo real são modelados como objetos. Mensagens são passadas entre eles, descrevendo ações a serem realizadas, enquanto que regras do tipo SE-ENTÃO, descrevem o comportamento que os objetos devam assumir.

Após o ROSS, outros sistemas incorporando idéias semelhantes tem aparecido. Dentre eles podemos citar o artigo de BENJAAFAR [1992]. Este trabalho relata alguns estudos e pesquisas para o desenvolvimento de um ambiente integrado de simulação com IA. A nova arquitetura proposta, esta baseada numa representação estruturada de objetos. O autor procura mostrar que a adoção desta técnica de construção de sistemas inteligentes de simulação trará, futuramente resultados mais efetivos e facilidades na integração entre modelagem, simulação e as funções de inteligência.

Outros trabalhos também podem ser citados: o SIMULATION CRAFT, da Universidade de Carnegie Mellon, o SIMKIT da Intellicorp e o LASER/SIM da Intellisys [TALAVAGE, 1987]. Estes sistemas incorporam estruturas de representação do conhecimento e modernas técnicas de POO.

2.8 Sumário

Neste capítulo abordamos a literatura relevante aos propósitos de nossa pesquisa. Enfatizamos, primeiramente, o conceito de flexibilidade e suas extensões aos sistemas de produção. Oportunamente, revisamos a literatura sobre os chamados SFM, definindo-os e discutindo seus principais elementos.

Uma vez que esta pesquisa pretende desenvolver um modelo para o tratamento de problemas de planejamento e controle da produção de SFM, procuramos revisar, de forma mais profunda, a literatura que trata destes problemas, subdividindo-os em seus principais segmentos afim de melhor compreendê-los e apresentá-los.

Antes de revisarmos a literatura que trata da aplicação de modelos de simulação ao planejamento e controle operacional de SFM, apresentamos e discutimos a avaliação de projetos de SFM, destacando os tipos de medidas e modelos utilizados.

Dentre os modelos focalizados no item anterior, a simulação tem se mostrado a ferramenta mais promissora para os fins a que nos propusemos. Desta forma, reservamos uma boa parte desta revisão à discussão da simulação aplicada à modelagem de SFM.

Encerramos a seção introduzindo algumas idéias recentes sobre a integração de ferramentas computacionais. Neste sentido os dois últimos temas revisados apontam para o uso e desenvolvimento dos chamados Sistemas Inteligentes de Simulação. Tais sistemas foram discutidos na última seção. Ressaltamos a significância e importância deste tipo de ferramenta, que combina as potencialidades da simulação e da inteligência artificial, no caso por meio dos Sistemas Especialistas. A propósito destes sistemas, antecedendo a discussão sobre os SIS, dedicamos uma seção inteira aos SE, definindo-os, discutindo sobre sua estrutura formal e sobre o uso de novas ferramentas para a sua construção.

CAPÍTULO III

MODELO DE SIMULAÇÃO COM ALOCAÇÃO DINÂMICA DE MÁQUINAS, PARA O PLANEJAMENTO E CONTROLE OPERACIONAL DE SFM

3.1 Planejamento da Produção Vs Controle Operacional em SFM

Conforme visto no item 2.3 da revisão bibliográfica, várias são as abordagens já experimentadas e adotadas para a solução do problema planejamento Vs operação de SFM. Dentre os mais promissores, estão os desenvolvidos por MAHESHWARI [1992] , GUPTA [1992], EGBELU [1993], YIM [1993] e ZHUANG [1993].

Os trabalhos de Gupta e Goyal GUPTA [1992] e de Maheshwari MAHESHWARI [1992], são por nós especialmente considerados. O primeiro por ser um trabalho de cunho investigativo bastante profundo, no qual os autores testaram inúmeras alternativas de gerenciamento da produção em SFM. Além do que, os testes foram realizados usando modelos de simulação, ferramenta de avaliação também por nós utilizada. O segundo trabalho não é apenas investigativo, como o de Gupta e Goyal. MAHESHWARI [1992] propõe, assim como nós, um modo integrado de abordagem do problema, por meio um processo iterativo, neste caso utilizando modelos de Programação Linear (PL) para a fase de planejamento, acompanhado de um modelo de simulação, no qual várias combinações de políticas operacionais são testadas, visando por em prática as soluções de planejamento determinadas pela PL. Os bons resultados reportados por estes trabalhos nos incentivou a investigar mais este tipo de abordagem.

Uma vez que os SFM tem suas operações quase que totalmente controladas por sistemas computacionais, em seus vários níveis hierárquicos, torna-se possível incorporar a estes sistemas controladores, algoritmos, que permitam a tomada de decisões, em tempo real, equivalentes àquelas provenientes dos processos de planejamento que adotem, por exemplo, modelos de PL.

Neste capítulo, propomos um novo modelo de simulação para SFM. Este modelo incorpora procedimentos de decisão especialmente voltados ao encaminhamento de soluções do problema de sequenciamento da produção de SFM. A união da simulação com algoritmos matemáticos, forma um sistema híbrido que busca soluções equivalentes àquelas tomadas em tempo real pelos sistemas hierárquicos de controle dos SFM.

Os procedimentos de decisão, alimentam o sistema com informações sobre o sequenciamento, objetivando, sobretudo, o balanceamento da carga das máquinas.

Vários dos elementos associados ao planejamento e controle operacional de SFM são utilizados no modelo desenvolvido e, por isso, são inicialmente revisados.

3.1.1 Elementos do Planejamento da Produção em SFM

Na figura 3.1 estão ilustrados os principais elementos envolvidos no processo de planejamento da produção de SFM.

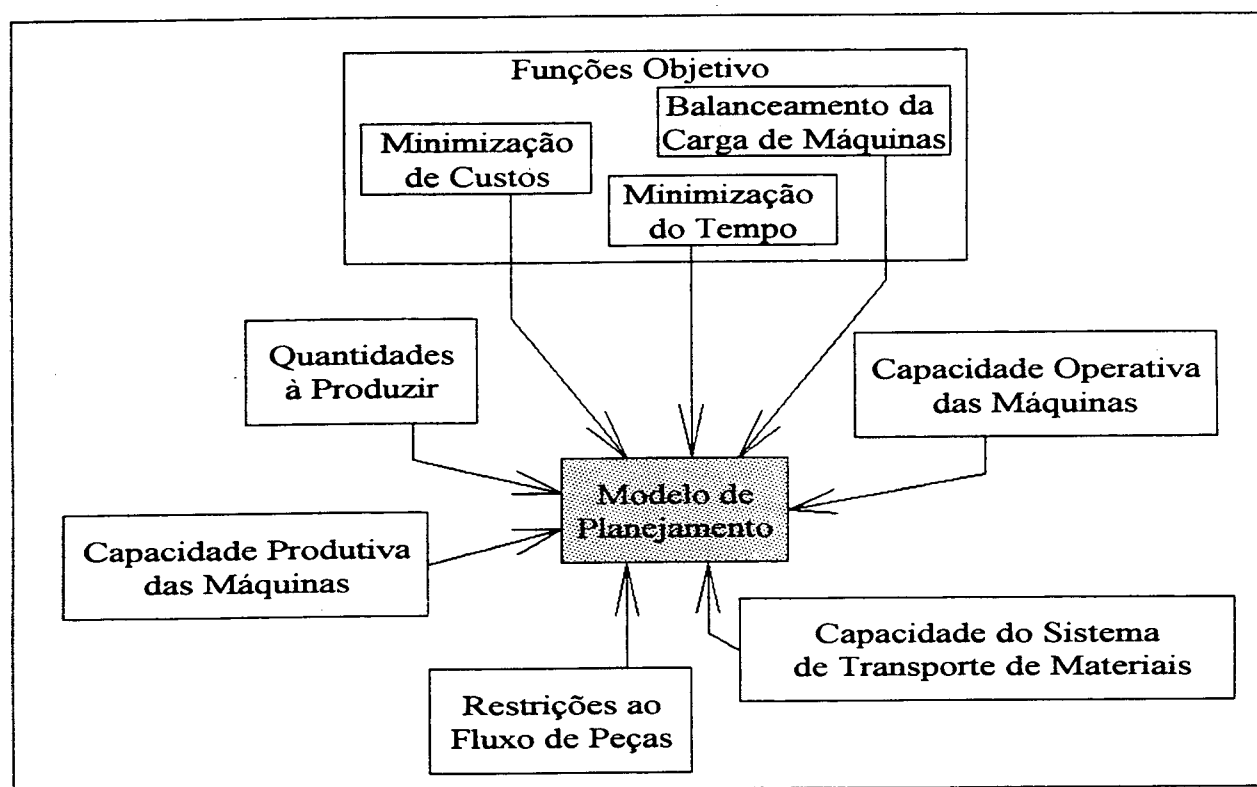


Fig. 3.1 - Elementos dos Modelos de Planejamento de SFM

Em seu mais baixo nível de decisão, o planejamento requer do decisor a definição da programação de todas as peças pertencentes ao lote de produção a ser manufaturado ao início de cada ciclo produtivo, que pode ser um turno, um dia, uma semana, etc.

Esta decisão, envolve vários parâmetros de planejamento tais como as necessidades de produção, por tipo de peça, bem como, as capacidades produtiva e operacional de cada centro de usinagem do sistema. Tais centros, embora com possibilidade de realizarem inúmeras operações, devido aos tipos de ferramentas existentes em seus carroceis de ferramentas, possuem capacidade limitada na execução dos diferentes tipos de operações.

Nos SFM, é grande o número de combinações que podem resultar em alternativas viáveis ao planejamento. Ao lado destas variáveis, o decisor deve ainda considerar a capacidade do sistema de transporte de materiais, elemento de fundamental importância nos SFM.

Paralelamente às necessidades de produção e às restrições do sistema, é necessário que o decisor determine um objetivo a ser alcançado com o planejamento. Vários objetivos diferentes são possíveis tais como:

- Minimização do custo operacional;
- Minimização da soma dos tempos de operação e transporte;
- Balanceamento da carga de cada centro de usinagem, etc..

A solução destes problemas de planejamento da produção, passa, em geral pela formulação de modelos de programação linear inteira, como os apresentados em STECKE [1986] e KUSIAK [1985]. A grande dificuldade porém, não se encontra nesta fase e sim quando da implementação operacional das soluções obtidas.

3.1.1.1 Hipótese Associadas aos Modelos de Planejamento de SFM

Na definição dos modelos de planejamento da produção de SFM a partir dos elementos ilustrados na fig. 3.3, as seguintes suposições são formuladas:

- as necessidades de produção de cada tipo de peça são conhecidas;

- é conhecido o plano de processo de cada tipo de peça, isto é, quais as operações e a ordem destas, necessárias a produção da peça ou componente;
- os tempos de processamento e de transporte são conhecidos e determinísticos;
- todos os custos relevantes podem ser obtidos;
- não são consideradas falhas instantâneas nas máquinas ou no sistema de transporte;
- o sistema se encontra vazio ao início de cada ciclo de manufatura;
- os VAGs utilizados possuem características similares.

3.1.2 Elementos do Controle Operacional de SFM

No item anterior apresentamos alguns dos principais elementos presentes nas decisões relacionadas com o planejamento da produção dos SFM. Entretanto, conforme visto nos capítulos anteriores, na etapa de execução muitas outras decisões de controle devem ser tomadas, as quais poderão afetar o planejamento e como consequência o próprio desempenho do sistema.

No nível de planejamento os modelos utilizados não incorporam muitas políticas operacionais com comprovada influência sobre o desempenho do sistema. Dentre as decisões operacionais de controle mais importantes temos:

- Políticas para a liberação de novas peças ao sistema;
- Políticas para a priorização de peças no buffer das máquinas;
- Políticas para o despacho de veículos transportadores.

Afora estas políticas operacionais, alguns recursos secundários do sistema, tais como o tamanho dos buffers e o nº de pallets, também não são incorporados aos modelos de planejamento embora também possuam grande influência sobre o desempenho operacional dos SFM [SCHRIBER, 1988; CO, 1988].

A figura 3.2 ilustra o controle operacional dos SFM e sua dependência das políticas operacionais, dos recursos secundários do sistema e das decisões tomadas na fase de planejamento.

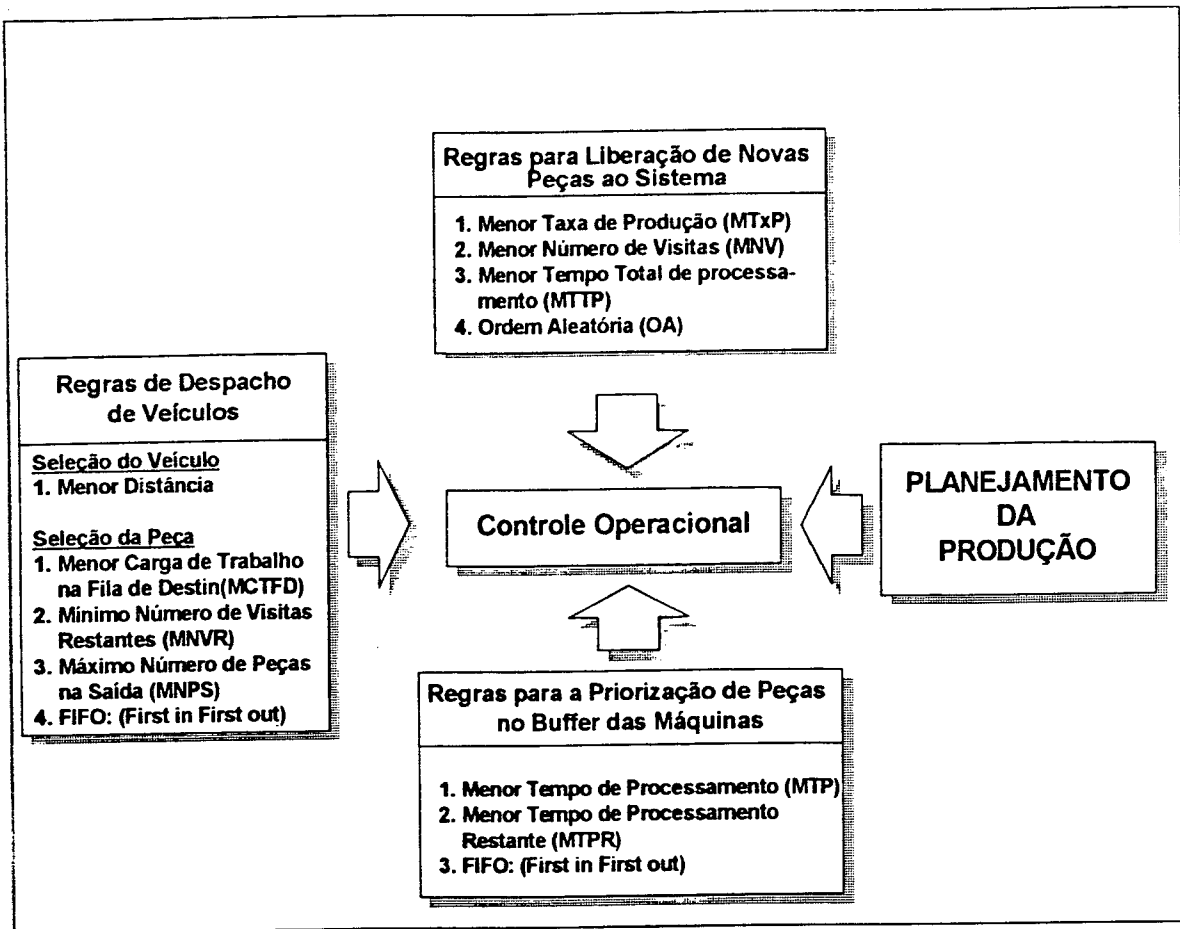


Fig. 3.2 - Elementos do Controle Operacional de SFM

Nos subitens que se seguem apresentamos uma descrição das políticas ou regras operacionais citadas acima, bem como dos chamados recursos secundários, buffers e pallets.

3.1.2.1 Regras para a Liberação de Novas Peças ao Sistema

Estas políticas ou regras, determinam, dentre as peças que estão aguardando para serem introduzidas no SFM, qual terá a maior a prioridade. Dentro destes sistemas existe um número limitado de peças circulando. Este número é controlado pelo número de pallets em circulação, uma vez que a peça se mantém num pallet enquanto no sistema. Um pallet torna-se disponível quando uma peça completa todas as suas operações e sai do sistema. Neste momento uma nova peça pode então ser introduzida no sistema. Uma regra de

prioridade deve ser definida para determinar qual dos diferentes tipos de peças que se encontram aguardando deve ser a escolhida. A literatura reporta vários trabalhos que tratam destas regras [NOF, 1979; CARRIE, 1985; WHITNEY, 1985; MONTAZERI, 1990; GUPTA, 1992].

Tais regras são, em geral, dependentes de características das peças, tais como, tempos de processamento, número de operações, tempo de chegada ou, do estado do sistema, que pode ser descrito, por exemplo, pela situação em que se encontram as máquinas que uma peça deve visitar ou pela taxa de produção do mix naquele momento. As regras mais utilizadas são:

- *Menor Taxa de Produção (MTxP)*: Esta regra depende do estado do sistema. É executado um acompanhamento, constante, da relação entre os tipos de peças já introduzidas no sistema, e o mix de produção desejado. É introduzida no sistema a peça com a menor relação. Esta regra procura manter o mix de produção desejado ao longo de todo o ciclo produtivo.
- *Menor Número de Visitas (MNV)*: Esta regra libera peças para o sistema, com base no número de visitas que uma peça tem que realizar para completar todas as suas operações, em uma ordem cíclica.
- *Menor Tempo Total de processamento (MTTP)*: Esta regra libera peças para o sistema, com base no tempo total de processamento necessário para que uma peça realize todas as suas operações, também numa ordem cíclica.
- *Ordem Aleatória (OA)*: Neste caso todas as peças aguardando possuem a mesma probabilidade de ser escolhida. Em geral este tipo de regra é adotado como forma de comparação no desempenho de outras regras

3.1.2.2 Regras para a Priorização de Peças no Buffer das Máquinas

Estas regras tratam da priorização das peças que se encontram nos buffers das máquinas aguardando para serem processadas. A prioridade é designada a uma peça, quando mais de uma se encontra aguardando. As regras mais citadas são:

- *Menor Tempo de Processamento (MTP)*: Esta regra prioriza aquela peça que possui o menor tempo de processamento para a realização da(s) operação(ões) na máquina..
- *Menor Tempo de Processamento Restante (MTPR)*: MTPR seleciona para processamento, a peça cujo tempo de processamento restante seja o menor.
- *FIFO: (First in First out)* : FIFO é utilizada como forma de comparar os desempenhos de regras alternativas. A prioridade é determinada de acordo com a ordem de chegada das peças nos buffers.

3.1.2.3 Políticas para o Despacho de Veículos Transportadores

As regras de despacho de veículos transportadores (VAGs), são necessárias quando uma peça precisa ser transportada de uma para outra máquina ou para a estação de carregamento/descarregamento. A prioridade deve ser designada quando mais de uma peça encontra-se aguardando por transporte no momento em que um veículo torna-se disponível. Estas regras podem ser dependentes das características das peças ou do estado do sistema [BOZER, 1989; LEUNG, 1990, GUPTA, 1992; MAHESHWARI, 1992; SABUNCUOGLU, 1992; YIM, 1993].

- *Menor Carga de Trabalho na Fila de Destino (MCTFD)*: A prioridade é dada à peça cuja fila de destino encontra-se com a menor carga de trabalho. A carga é medida pela soma dos tempos de processamento das peças que se encontram naquela fila. O objetivo desta regra é reduzir o desbalanceamento na carga das máquinas.
- *Mínimo Número de Visitas Restantes (MNVR)*: Por esta regra a prioridade é designada de acordo com o número de visitas que restam a uma peça, para que esta complete seu processamento. Por esta regra, as peças com maior valor agregado (mais operações completadas), terão maior prioridade no transporte
- *Máximo Número de Peças Esperando para Sair (MNPS)*: Esta regra determina que a prioridade deva ser dada a peça que se encontrar numa máquina, cuja fila de peças esperando para serem transportadas a outra estação, seja a maior.

- *FIFO: (First in First out)* : FIFO é utilizada como forma de comparar os desempenhos de regras alternativas. A prioridade é determinada de acordo com a ordem de solicitação de transporte das peças.

3.1.3 Parâmetros Secundários do Sistema

Em conjunto com as estratégias de controle operacional do sistema, existem alguns parâmetros operacionais tais como o tamanho dos buffers e o número de pallets que possuem comprovado impacto sobre o desempenho do sistema [CO, 1988; SCHRIER, 1988; GUPTA, 1992; MAHESHWARI, 1992; SABUNCUOGLU, 1992]. Estes dois elementos são elementos fundamentais desta investigação sendo descritos com mais detalhes na apresentação do modelo de simulação desenvolvido.

3.2 Modelo para o Planejamento e Controle Operacional de SFM

No item 2.4.2 da revisão bibliográfica (Modelos para a Avaliação de SFM), discutimos sobre a utilidade de modelos analíticos no estudo dos aspectos operacionais de SFM reais. Tais modelos tem pouca utilidade neste aspecto, por não suportarem a incorporação dos princípios que governam a dinâmica operacional destes sistemas de manufatura. Por isso a adoção, neste trabalho, de um modelo de simulação discreta. Os modelos de simulação possuem a capacidade de emular, com muita similaridade, modelos reais de SFM, embora, ressalte-se, sejam modelos tipicamente avaliativos, não permitindo otimizações. [SURI, 1985]. Em outras palavras, um modelo de simulação requer que algumas decisões sejam tomadas "a priori", gerando alternativas que possam ser avaliadas num ambiente dinâmico e detalhado como os fornecidos pelos modelos de simulação.

O modelo de simulação foi desenvolvido usando-se a versão estudantil da linguagem de simulação SIMAN 4.0 [PENG DEN, 1990]. Esta decisão deveu-se, em parte, a excepcional capacidade desta linguagem em simular sistemas de produção, principalmente àqueles com características especiais de sistemas de transportes de materiais, como os SFM. O segundo motivo da adoção do SIMAN, foi sua disponibilidade, à época, nos laboratórios da University of South Florida (USF), onde os modelos foram, inicialmente, estudados e desenvolvidos.

O modelo de simulação consiste na integração de três grandes circuitos lógicos de controle: o controle das máquinas, o controle dos veículos transportadores e o controle do fluxo das peças. Pelo fato de não utilizarmos uma estratégia comum para o controle do fluxo das peças e por tratar-se de uma versão estudantil da linguagem de simulação, fomos obrigados a desenvolver parte do modelo na linguagem de programação C, conectando-se os módulos assim desenvolvidos ao modelo em SIMAN.

Um aspecto importante na modelagem voltada à simulação é a verificação e validação dos modelos. Os textos sobre simulação [BANKS, 1984; LAW, 1991; PENGDEN, 1990], definem verificação como sendo o processo de determinar que o modelo funciona conforme o projetado, isto é, isento de erros de programação. Já a validação, consiste em determinar se o modelo é uma representação acurada do sistema real sob estudo. Nesta pesquisa, não será possível uma plena validação, uma vez que os modelos são hipotéticos, isto é, não são baseados em sistemas reais. No entanto, os sistemas modelados foram baseados em sistemas já utilizados em pesquisas anteriores [CARRIE, 1988; MAHESHWARI, 1992]. A validação, neste caso consistiu na reprodução dos resultados daquelas pesquisas, quando parâmetros de entrada semelhantes foram utilizados.

Quanto à verificação, ambos os programas, em SIMAN e em C, foram exaustivamente testados afim de se tornarem isentos de erros de programação. Sua lógica foi "rastreada", usando-se as facilidades do "Trace Element" do SIMAN. Os fluxos das entidades dentro do modelo foram acompanhados em inúmeros testes, afim de assegurar sua lógica.

3.2.1 Parâmetros Utilizados no Modelo

Seleção de Peças: Uma vez instalado, o SFM opera para produzir, simultaneamente, uma variedade de peças em pequenos lotes. Esta variedade de peças se restringe ao mix para o qual o sistema foi projetado.

Na literatura encontra-se a descrição de várias técnicas utilizadas para a seleção de peças. Técnicas baseadas em tecnologia de grupo [BROWNE, 1984] ou em algoritmos baseados em programação linear inteira [WHITNEY, 1985], são dois exemplos. No entanto, para os propósitos desta pesquisa, é suficiente reconhecer que os SFM podem produzir peças de acordo com seus projetos, com base em algum tipo de atributo de manufatura, e que o número de tipos de peças pode ser muito grande. Neste estudo, é possível testar-se o

modelo e o protótipo com uma pequena parte desta grande família de peças. Na literatura verificamos que o número médio de variedade de peças produzidas é 20 [BROWNE, 1984; BUZACOTT, 1986a e 1986b]. Este número será adotado neste trabalho.

Número de Operações: Cada peça necessita um determinado número de operações para completar seu processamento. Neste trabalho o nº de operações por peça é assumido como sendo uniformemente distribuído, assumindo valores discretos entre 3 e 10, com uma média de 6,5 operações/peça.

Tempo de Processamento: Não existe na bibliografia revisada, relatos específicos a respeito de tempos de processamento nas peças em SFM. Tais tempos são correspondentes àqueles decorridos em processos semelhantes nos sistemas convencionais. Neste estudo, assumimos que os tempos são determinísticos e podem variar de 2 min. à 20 min. O tempo de processamento em máquinas alternativas, isto é, naquelas com menor eficiência no processamento de determinadas operações, sofrerão acréscimos que variam de 20% à 100%, do tempo das máquinas principais.

Mix de Peças: Num determinado ciclo de produção, somente um certo número de tipos de peças é selecionado para ser produzido, obedecendo a uma determinada relação. Assumimos que os tipos de peças variam entre um mínimo de 3 e um máximo de 8, num ciclo. Cada peça selecionada, terá uma participação relativa de 1, 1.5, 2.0, 2.5 ou 3.0. Desta forma, se um tipo de peça recebe peso 1 e outra peso 1.5, a segunda terá uma participação 50% maior que a primeira no mix de produção. A seleção dos pesos será baseada na distribuição uniforme discreta.

Número e Velocidade de VAGs: A decisão sobre o número de Veículos Auto Guiados a serem utilizados em um SFM, objetivando assegurar uma particular taxa de produção, depende, dentre outras variáveis, do arranjo físico do sistema e de sua programação. Esta decisão é típica da fase de detalhamento do projeto, na qual um nº razoável de alternativas de programação da produção devem ser testadas (simuladas) em relação a um determinado arranjo físico, para que se possa concluir algo, a partir de inferências sobre os resultados do modelo testado. Na literatura [EGBELU, 1984; RAMAN, 1989; SABUNCUOGLU, 1992; YIM, 1993], verificamos que um número típico de VAGs, situa-se entre 1 e 4 veículos, dependendo do sistema. Trabalhamos, neste estudo com estes números. Quanto a velocidade, o mais comum são velocidades nominais médias da ordem de 30 m/s à 50

m/s. Ambas as variáveis, número e velocidade, podem ser facilmente alteradas pelo analista durante os procedimentos de avaliação do desempenho dos sistemas.

Número de Pallets e Tamanho dos Buffers: Pallets são considerados peças chaves no desempenho operacional de SFM e, a decisão sobre que número de pallets utilizar, depende, basicamente, da disponibilidade de espaço nos buffers das máquinas. O número de pallets no sistema, obviamente não pode ser maior que o tamanho dos buffers, que nos SFM são bastante restritos, resumindo-se à pequenos espaços em frente as máquinas e no próprio sistema de transporte. No caso de um controle falho, a ocorrência de uma situação como essa, levaria ao bloqueio do sistema.

Com base nos trabalhos de Schriber e Stecke [SCHRIBER, 1988], MAHESHWARI [1992]; SABUNCUOGLU [1992] e YIM [1993] assumimos o mesmo tamanho de buffer em cada máquina. cinco diferentes capacidades, 2, 3, 4, 5 ou 6, são adotadas.

Por outro lado, o número de pallets não deve ser menor que o número de máquinas, o que causaria uma baixa utilização do sistema, pois nesse caso, em qualquer momento alguma máquina pode estar em desuso por falta de pallets. Usualmente, o número de pallets e o número de máquinas, relacionam-se com índices de 1.5, 2.0, 2.5, 3 ou 4 vezes, isto é, num sistema com 4 máquinas pode-se adotar 6, 8, 10, 12 ou 16 pallets

3.2.2 Modelo de Simulação Proposto

Os parâmetros básicos de entrada do modelo são os definidos no item anterior. Para que o sistema possa funcionar é preciso, no entanto, que para cada tipo de peça a ser processada exista uma sequência de operações previamente determinada, a qual está, normalmente, atrelada a uma sequência de visitação de máquinas, de tal forma que, no momento de sua entrada no sistema, seja possível determinar seu destino.

No modelo proposto, a determinação da sequência de visitação das máquinas funciona de maneira diferente, das propostas nos modelos que classificamos como tradicionais. Nestes, a sequência é estabelecida *a priori*, com auxílio ou não de modelos matemáticos.

Adotamos aqui uma sequência de visitas flexível, isto é, com a possibilidade de mais de uma máquina poder realizar uma mesma operação, embora com diferentes eficiências,

sobre uma peça. Será o estado do sistema que determinará qual máquina, naquele momento, deverá ser alocada para a execução do trabalho.

Para que isto seja possível, um algoritmo programado em C e conectado ao modelo em SIMAN, é executado sempre que exista a necessidade de execução de uma nova operação em uma peça, uma vez que, neste momento, uma decisão sobre qual máquina realizará a operação deve ser tomada. A lógica deste controle é apresentada no item 3.2.2.1.

A simulação será iniciada com o sistema vazio e todos os equipamentos disponíveis. Inicialmente, o número de peças liberadas ao sistema é igual ao número de pallets. Subseqüentemente, as peças entram no sistema à medida em que os pallets vão ficando disponíveis. As peças são liberadas em acordo com a política de liberação adotada. Após completar todo seu processamento, as peças são transportadas de volta a estação de Carregamento/Descarregamento e o respectivo pallet fica disponível.

O tempo de execução de uma simulação depende, usualmente, dos objetivos do analista. Na grande maioria dos processamentos ligados a projetos de manufatura, este tempo está associado ao ciclo de produção de um determinado conjunto de peças. Outras vezes, este tempo relaciona-se com os turnos de produção do próprio sistema, como, por exemplo, quando se deseja saber qual seria a produção alcançada, no período de uma semana, com turnos diários de 8 horas.

Nos exemplos que serão aqui apresentados, a execução das simulações se encerrarão no momento em que todas as peças requeridas em um determinado ciclo produtivo forem manufaturadas. Isto só não ocorrerá, quando do acontecimento de um evento extraordinário, evitando a concretização de um ciclo.

Um exemplo de evento extraordinário, poderia ser o bloqueio do sistema devido a falhas no projeto ou no planejamento e controle da produção. Portanto, o período de execução de uma simulação é uma variável e é igual ao *Tempo de um Ciclo de Produção*. Esta variável é importante quando se deseja comparar sistemas produtivos diferentes ou mesmo diferentes estratégias de produção sobre um mesmo sistema

3.2.2.1 Controle Lógico da Sequência de Visitações de Máquinas

Conforme comentamos no item anterior, no modelo de simulação desenvolvido, não adotamos uma sequência pré-programada de visitação de máquinas pelas peças. Em substituição a este procedimento, geralmente voltado ao emprego de modelos analíticos, como a programação linear, introduzimos a idéia de definir as rotas das peças que se encontrem no sistema, no momento exato em que esta decisão deva ser tomada e não previamente.

Tal idéia procura buscar uma solução que permita utilizar as características de flexibilidade dos SFM ditadas, principalmente, pela multiplicidade operacional de seus centros de usinagem e, ao mesmo tempo, evitar o descompasso entre o planejamento e a operação devido as dificuldades dos modelos matemáticos em incorporarem o dinamismo operacional destes sistemas.

A lógica por traz desta idéia é simples. Sempre que houver a necessidade de executar uma nova operação em uma peça, o programa executa uma pesquisa sobre os centros de usinagem ou máquinas, de tal forma que obtenha como resposta, qual máquina deverá efetuar a operação. Nesta busca, são considerados os seguintes elementos:

- máquinas que possuem a capacidade de executar a operação desejada;
- eficiência na execução da operação;
- carga de trabalho associada às máquinas candidatas.

Os dois primeiros elementos são parâmetros das máquinas do sistema, enquanto que o terceiro é determinado pelo estado do sistema no momento da decisão.

Na figura 3.3., a seguir, apresenta-se um fluxograma do algoritmo proposto. No Anexo 01 apresenta-se seu código.

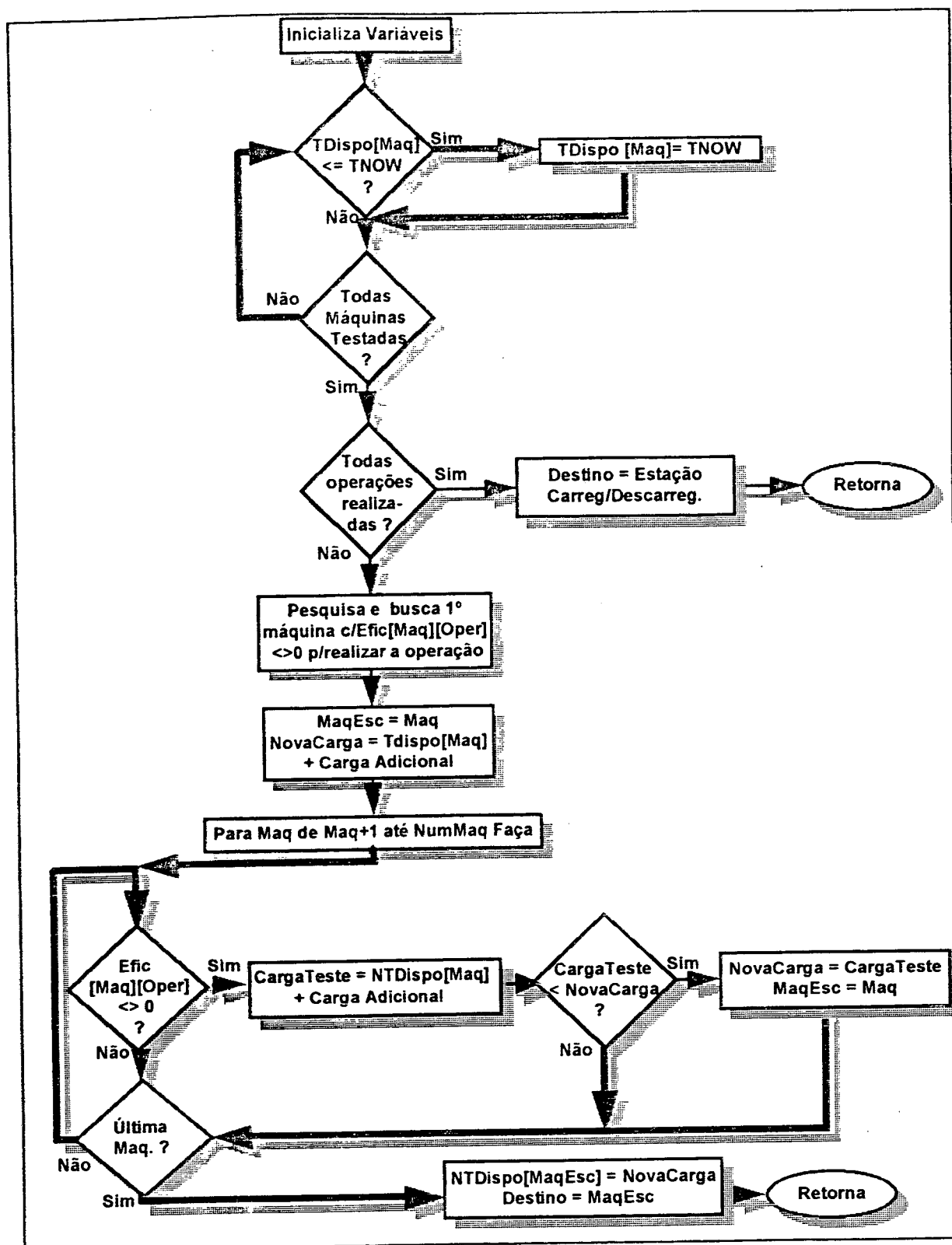


Fig. 3.3 Controle Lógico da Sequência de Visitações de Máquinas

Abaixo, apresenta-se passo-a-passo os procedimentos efetuados pelo programa.

Passo 1. Inicialização

Inicializa todas as variáveis do processo.

Passo 2. Atualização do Tempo Disponível das Máquinas

Verifica, para todas as máquinas, o valor de TDispo, isto é, o tempo em que a máquina estará disponível após realizar toda a carga de trabalho a ela designada. Se a máquina estiver liberada, TDispo será menor ou igual ao atual tempo de simulação TNOW. Neste caso, faz $TDispo = TNOW$.

Passo 3. Verificação do Tipo de Operação

Verifica o tipo de operação a ser realizada. Se TipOper = -1, todas as operações já foram realizadas sobre a peça. Neste caso encaminha a peça a estação de Carregamento/Descarregamento e retorna ao programa de simulação. Caso contrário segue para o passo 4.

Passo 4. Inicialização do Processo de Busca da Melhor Máquina

Busca a primeira máquina com capacidade de realizar a operação desejada, isto é, $Efic [Maq] [TipOper] > 0$. Faz $MaqEsc = Maq$.

Passo 5. Cálculo da Carga Adicional e Inicialização do Melhor Tempo Disponível

Calcula a carga adicional à carga da máquina escolhida MaqEsc. Neste cálculo considera o tipo da peça, a operação a ser realizada e a eficiência da máquina MaqEsc. Em função da carga adicional, calcula o novo tempo disponível para a máquina MaqEsc. Faz $NovaCarga = TDispo[Maq] + CargaAdicional$

Passo 6. Busca do Melhor Tempo e Melhor Máquina

Busca a próxima máquina que possa realizar a operação desejada, isto é, àquela com eficiência positiva para a referida operação. Calcula a sua carga adicional e seu novo tempo disponível. Atribui o resultado a CargaTeste

Passo 7. Definição da Melhor Máquina

Testa se $CargaTeste$ calculado é menor que $NovaCarga$. Se verdade, faz $NovaCarga = CargaTeste$ e $MaqEsc = Maq$.

Passo 8. Teste de Parada..

Se todas as máquinas com eficiência positiva relativas a operação desejada já foram testadas, segue para o passo 9. Senão volta ao passo 6

Passo 9. Final de Procedimento.

Faz $TDispo [MaqEsc] = NovaCarga$. Faz a compatibilização entre as variáveis do programa em C e do programa de simulação em SIMAN. Retorna ao programa de simulação

3.2.2.2 Hipóteses Associadas ao Modelo

Os modelos de simulação, em geral, não requerem tantas hipóteses quanto os modelos analíticos. No entanto, o desenvolvimento de qualquer tipo de modelo, necessita um claro delineamento de suas fronteiras. Além disso, pode não ser apropriado, nem mesmo viável, incorporar todo o tipo de detalhamento ao modelo de simulação. Assim, para manter o esforço de modelagem sobre controle é necessário fazermos algumas hipóteses razoáveis sobre o sistema. No modelo desenvolvido, as seguintes hipóteses foram feitas:

- Todos os pallets encontram-se, inicialmente, na estação de Carregamento.
- Todos os VAGs, encontram-se, inicialmente, na estação de VAGs.
- O nº de pallets disponível é fixo para um determinado ciclo de produção.
- Todas as matérias primas das peças a serem produzidas num dado ciclo encontram-se, inicialmente na estação de Carregamento.
- As estratégias de controle operacional são pré-definidas e não mudam durante o ciclo de produção.
- Os VAGs retornam a sua estação se a fila de requisição de transporte estiver vazia.
- Uma peça só é movida para outra estação se houver espaço disponível para seu pallet.

- O controle de decisões sobre os VAGs só é efetuado junto as máquinas, estação de VAGs e estação de Carregamento/Descarregamento.
- O controle de destino de uma peça só é efetuado junto às máquinas ou estação de Carregamento/Descarregamento.

3.2.2.3 Controle Lógico do Fluxo de Peças

O fluxo lógico de peças e pallets é o mesmo. Uma peça só é liberada para ser processada quando um pallet está disponível. Logo, os fluxos de peças e pallets são análogos no modelo de simulação. O número simultâneo de peças no sistema é limitado pelo número de pallets. A liberação de peça ao sistema é feita de acordo com as *regras de liberação de peças*. O controle do fluxo das peças no sistema, é feito pela sequência de operações, sem contudo estar associado, especificamente, às máquinas, existindo, portanto, uma flexibilidade de rotas.

Uma peça pode encontrar-se em qualquer uma das quatro seguintes condições no sistema modelado:

- sendo processada em uma máquina,
- esperando por processamento em uma área de armazenagem junto as máquinas,
- aguardando por transporte ou
- em transito para seu próximo destino.

Para minimizar a possibilidade de bloqueio do sistema, uma peça só é transportada para uma próxima máquina se houver espaço em sua área de armazenagem.

A figura 3.4 ilustra o controle de fluxo de peças.

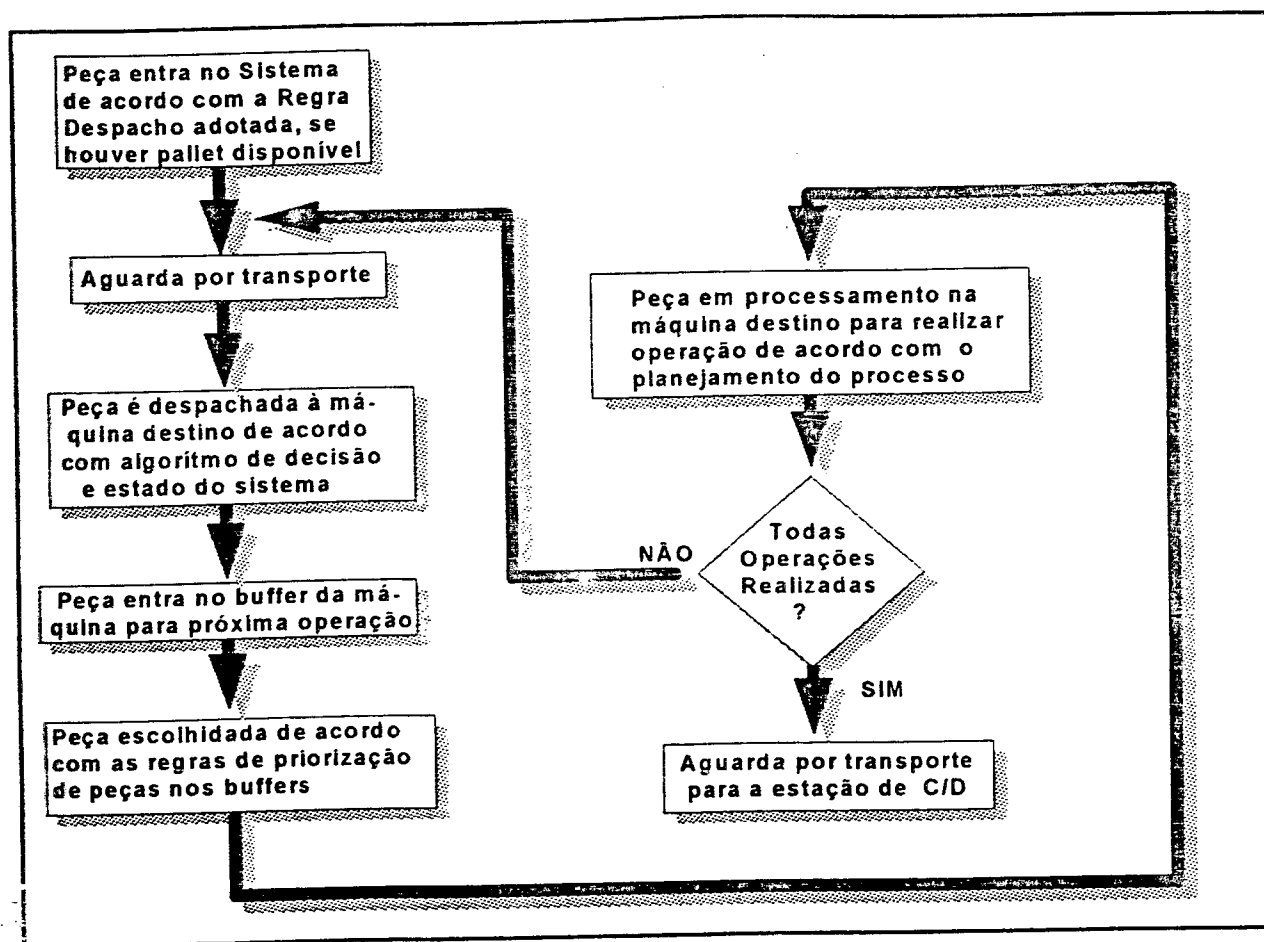


Fig. 3.4: Controle do Fluxo de Peças

3.2.2.4 Controle Lógico das Máquinas

O controle lógico das máquinas, está totalmente associado às peças que podem estar sendo processadas ou aguardando, para serem processadas ou transportadas. Se mais de uma peça aguarda por processamento, então a regra de prioridade de sequenciamento é aplicada para selecionar uma peça. Empates são decididos com base no tempo de espera na fila. A fig. 3.5 ilustra esta lógica.

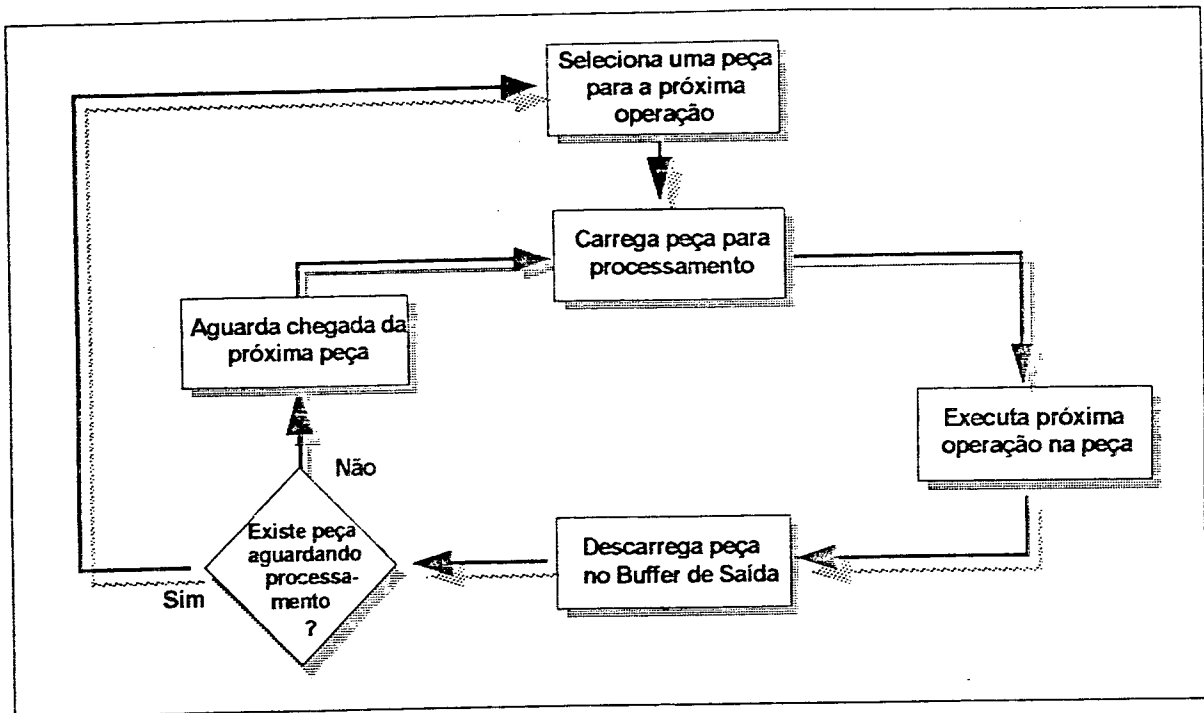


Fig. 3.5: Controle Lógico das Máquinas

3.2.2.5 Controle Lógico do Fluxo dos VAGs

Dentro do modelo de SFM, o transporte é executado por Veículos Auto Guiados que merecem um controle especial sobre seus movimentos. Estes veículos, podem se movimentar sobre um caminho pré-determinado para alcançarem qualquer uma das máquinas, sua própria estação de permanência ou a estação de Carregamento/Descarregamento.

O caminho sobre o qual os veículos se movimentam, foi dividido em pequenos segmentos, os quais são controlados pelos veículos na medida de suas movimentações. Cada segmento é assumido como se fora um recurso, que é tomado e liberado pelo veículo quando de sua passagem sobre ele. Sendo assim, dois veículos não podem compartilhar o mesmo recurso e portanto não podem estar no mesmo lugar ao mesmo tempo. Da mesma forma, somente um veículo pode encontrar-se junto a área de armazenagem das máquinas. Com exceção da estação de Carregamento/Descarregamento, área das máquinas e estação dos VAGs, que são caminhos bi-direcionais, todos os outros são unidirecionais. A velocidade é assumida como constante durante a execução de uma simulação e, é cerca de 30% menor nos segmentos bi-direcionais e curvas.

Todas as prioridades sobre o transporte das peças são determinadas pelas regras de despacho e de acordo com as hipóteses gerais associadas ao modelo.

A lógica do fluxo dos veículos é apresentada na figura 3.6.

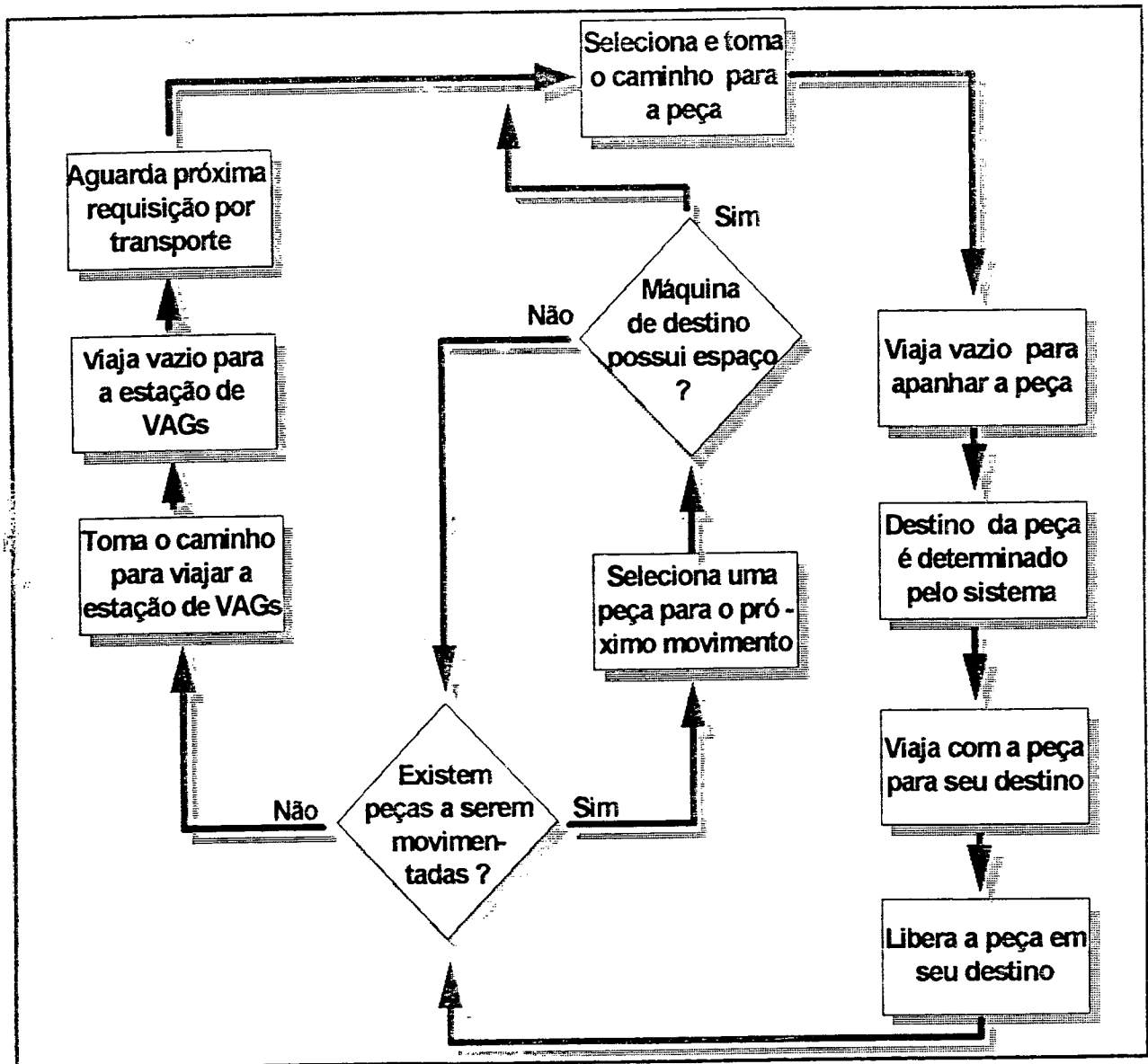


Fig. 3.6: Controle Lógico do Fluxo de VAGs

3.3. Medidas de Desempenho

As medidas de desempenho operacional para a avaliação simulada de sistemas de produção podem ser as mais diversas. As principais medidas de desempenho operacional utilizadas neste estudo são:

- o tempo de passagem;
- o tempo do ciclo produtivo;
- a taxa de utilização das máquinas;
- a taxa de utilização dos VAGs;
- o tamanho das filas nas máquinas

O tempo de passagem é definido como o tempo que uma peça permanece no sistema, iniciando-se sua contagem no momento em que esta é liberada para o sistema na estação de Carregamento/Descarregamento, mais precisamente, quando a peça é carregada em um pallet. A contagem deste tempo encerra-se no exato momento em que a peça é descarregada do pallet, na estação de Carregamento/Descarregamento, saindo do sistema.

O tempo do ciclo de produção, é definido pelo tempo total para a manufatura de todas as peças requeridas. Nos ambientes produtivos sob investigação, somente um número finito de peças é produzido em cada ciclo. Da literatura [MAHESHWARI, 1992], fica comprovado que medir o tempo do ciclo produtivo, em um SFM, é análogo a medir-se a utilização do sistema como um todo ou sua taxa de produção. Portanto, as duas últimas medidas não são usadas neste trabalho.

A taxa de utilização das máquinas é determinada pela relação entre o tempo que cada máquina passa efetivamente processando peças e o tempo total do ciclo produtivo.

A taxa de utilização dos VAGs é determinada pela relação entre o tempo que cada VAG passa movimentando-se, carregado ou vazio e o tempo total do ciclo produtivo. Outra taxa que pode ser utilizada é a de utilização efetiva dos VAGs, na qual somente o tempo de uso com carga é computado.

O tamanho das filas das máquinas é fornecido como resultado de uma média ponderada pelo tempo, do número de elementos que se encontrem, ao longo da simulação, naquelas filas. Embora com capacidade limitada pelo tamanho dos buffers, é uma importante medida, diretamente relacionada com detecção de gargalos.

3.4 Sumário

Neste capítulo apresentamos os principais elementos envolvidos no problema do planejamento e controle operacional dos SFM. Apresentamos, também, um modelo de simulação desenvolvido para tratar estes problemas. Neste modelo, uma atenção especial foi dada a fase de planejamento, pela integração de um algoritmo para executar o controle sobre a sequência de visitação de máquinas. Este algoritmo, torna desnecessária a alocação prévia de recursos, usualmente executada por modelos de programação linear inteira ou modelos de programação por múltiplos objetivos e, cujos resultados costumam causar problemas na fase operacional.

No capítulo que segue, trataremos da aplicação do modelo aqui desenvolvido a uma série de problemas testes. Será feito, também, um teste comparativo com uma aplicação que faz uso de um modelo de programação linear para a fase de planejamento.

CAPÍTULO IV

APLICAÇÃO DO MODELO PROPOSTO E COMPARAÇÃO DOS RESULTADOS COM OS DOS MODELOS TRADICIONAIS

Neste capítulo apresentamos os resultados práticos da aplicação da abordagem proposta e do modelo de simulação dela derivado a problemas de planejamento e controle operacional de SFM. Esta aplicação tem como objetivos, primeiro a validação do modelo desenvolvido e, segundo, a demonstração de sua contribuição a solução do problema em questão. Para tal, verificamos seu comportamento em comparação com resultados obtidos por outros pesquisadores que trataram o mesmo tipo de problema utilizando diferentes abordagens.

Para a obtenção dos resultados práticos são necessários um SFM e um conjunto de problemas exemplos que permitam a validação desejada. Dois trabalhos obtidos na literatura proporcionam estes elementos. O SFM utilizado, foi apresentado e modelado por CARRIE [1988], e baseia-se em um sistema real (ver figura 4.1).

Dois motivos nos levaram à escolha deste sistema em particular. O primeiro, é o fato de estar bem documentado na literatura, o que possibilita a validação da modelagem por nós desenvolvida, uma vez que se pode repetir os resultados obtidos por Carrie quando os mesmos parâmetros de entrada são utilizados. O segundo motivo é ter sido este sistema também utilizado por MAHESHWARI [1992] em um importante trabalho de pesquisa, já anteriormente citado, e que trata justamente do problema da compatibilização entre as ações de planejamento e operação em SFM.

Desta forma, este capítulo está dividido em seções com dois propósitos distintos. Nas duas primeiras, 4.1 e 4.2, apresentamos o SFM a ser modelado, e os testes de validação do modelo desenvolvido. Na seção seguinte, 4.3, tratamos da aplicação da abordagem desenvolvida, em problemas de planejamento e controle da produção sobre o SFM descrito no item 4.2.. Mostramos os resultados obtidos e os comparamos com aqueles

apresentados por Maheshwari, que utilizou-se de uma metodologia envolvendo Programação Linear e Simulação.

4.1 Características do Sistema Flexível de Manufatura Modelado

O SFM que servirá como exemplo (ver layout na figura 4.1 abaixo), é o mesmo tratado e apresentado por CARRIE [1988]. O sistema é constituído de quatro centros de usinagem, não-idênticos, e ferramentas instaladas em magazines com capacidade para 50 diferentes tipos. Além das ferramentas, tais máquinas, diferenciam-se em outros aspectos, tais como: o número de eixos do controle automático, o número de eixos de rotação, potência disponível, etc. As várias operações necessárias a manufatura de peças pertencentes as diferentes famílias, podem ser efetuadas em múltiplos centros de usinagem. No entanto, em função de suas características próprias e das ferramentas neles instaladas, os centros possuem diferentes eficiências na execução das mesmas operações.

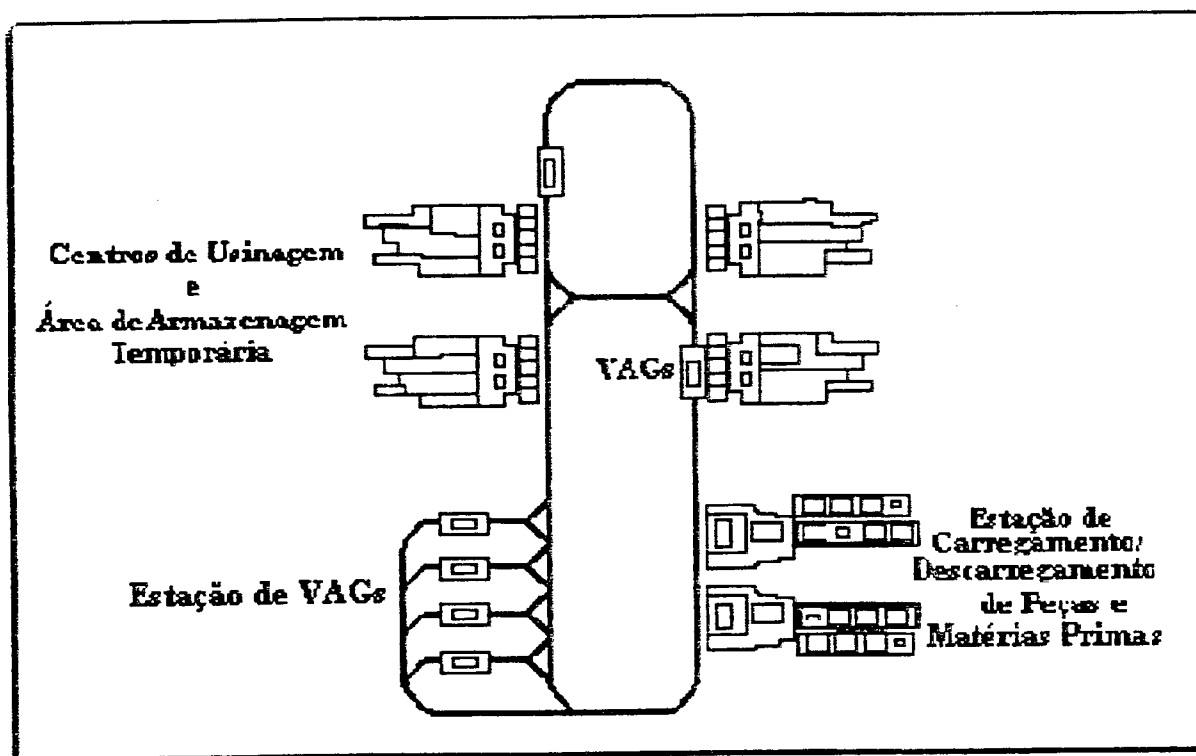


Figura 4.1: Layout do SFM exemplo baseado no modelo citado por CARRIE [1988]

Apesar destas diferenças nas eficiências, o sistema possui grande flexibilidade, principalmente de rotas, considerada a mais importante quando se trata do planejamento da produção no curto prazo.

Em função dos objetivos de flexibilidade desejados, o sistema de transporte de materiais adotado é constituído de Veículos Auto Guiados, VAGs, cujo número é considerado variável e determinado de acordo com as avaliações dos resultados das simulações. Cada estação de trabalho possui um buffer, cujo tamanho também é objeto das decisões a serem tomadas quando da aplicação do modelo.

4.2 Validação do Sistema Flexível de Manufatura Modelado

O modelo de SFM apresentado no item 4.1, foi modelado utilizando-se uma linguagem de simulação (SIMAN), diferente daquela originalmente empregada por Carrie [1988] (ECSL). SIMAN e ECSL são linguagens apropriadas para duas diferentes formas de abordagem e modelagem de sistemas. A linguagem ECSL empregada por Carrie, é voltada para a abordagem por eventos, empregando o método das três fases, típica da escola inglesa. Já a linguagem SIMAN, é voltada, principalmente para uma abordagem por processos (escola americana), embora permita também, por meio de ligações com rotinas externas, a abordagem por eventos, se necessário. Além das diferenças já citadas, existe ainda o fato de que dois modelos de simulação, criados por dois diferentes programadores, empregando a mesma linguagem, raramente serão iguais em vista da parcela de arte envolvida em seu desenvolvimento. Desta forma, uma vez que vamos utilizar o mesmo SFM para demonstrar a metodologia desenvolvida, é necessário validar o modelo desenvolvido em SIMAN.

Para avaliação do sistema modelado, simulamos o mesmo problema resolvido por Carrie, utilizando os mesmos parâmetros e dados de entrada. Repetimos a simulação com dez diferentes valores de sementes geradoras de valores aleatórios e comparamos os valores médios obtidos das simulações com aqueles gerados por Carrie. Um teste de hipóteses para as médias dos tempos de ciclo de produção e dos tempos de passagem foi realizado e seus resultados mostraram a validade do modelo programado em SIMAN. A seguir apresentamos o problema proposto, os resultados das simulações e os resultados dos testes de hipóteses.

4.2.1 Problema Utilizado na Validação

O modelo de SFM apresentado por CARRIE [1988], propõe a simulação de uma semana de produção, com turnos de 8 horas, de cinco diferentes tipos de peças, aqui tratadas como tipos 1, 2, 3, 4 e 5.

O mix desejado para as peças é apresentado na tabela 4.1.

Tipo de Peça	1	2	3	4	5
Participação	25%	20%	15%	18%	22%

Tabela 4.1: Mix de Produção Desejado

De acordo com seus respectivos planos de processo, as seqüências de operações são designadas aos centros de usinagem existentes no sistema. A tabela 4.2. apresenta, para cada tipo de peça do mix, a sua seqüência de operações, indicando a máquina na qual a operação será realizada e o respectivo tempo em minutos.

Por exemplo, a peça tipo 1, necessita passar por 6 diferentes operações, as quais deverão ser realizadas nas máquinas 4, 1, 2, 1, 3 e 1, respectivamente. Os tempos das operações em cada máquina, de acordo com a tabela serão de 19, 15, 14, 11, 8 e 4 min. respectivamente.

TIPO de PEÇA	OPERAÇÕES (Máquinas/Tempos (min))								
	1º	2º	3º	4º	5º	6º	7º	8º	9º
1	4/19	1/15	2/14	1/11	3/8	1/4	-	-	-
2	1/11	4/2	3/15	1/9	3/6	2/13	3/6	4/11	1/19
3	3/16	2/2	3/8	1/5	3/9	1/17			
4	3/8	2/7	4/7	2/15	3/13	4/7	2/4		
5	2/10	4/6	1/7	4/10	1/12	2/18	3/10	1/17	4/17

Tabela 4.2: Seqüências Operacionais e Respectivos Tempos

4.2.2 Os Resultados Obtidos

O problema proposto foi simulado dez vezes, com diferentes sementes geradoras dos valores aleatórios gerando dez experimentos. Os resultados, obtidos pelo modelo programado em SIMAN, são apresentados na tabela 4.3.

Nº do Teste	Produção p/Tipo (unid)					Total	Tempo Médio Fluxo p/Tipo (min)					Total
	1	2	3	4	5		1	2	3	4	5	
1	34	25	19	20	22	120	239	291	202	182	309	247
2	23	26	19	17	29	114	279	336	228	202	336	287
3	29	20	12	23	30	114	209	296	185	176	309	241
4	25	24	18	20	29	116	252	323	218	196	361	279
5	32	25	12	21	28	118	254	337	227	214	340	282
6	25	26	17	33	19	120	210	273	204	176	289	226
7	27	34	17	11	20	109	358	412	303	241	418	365
8	27	27	16	27	24	121	234	318	191	194	308	253
9	23	17	16	25	33	114	273	335	216	189	322	270
10	29	30	17	18	24	118	279	327	226	206	353	288
Médias	27,5	25,6	16,6	21,9	26,3	116,4	258,8	325,0	220,3	198,0	335,0	273,8

Tabela 4.3: Resultados dos dez experimentos com o modelo em SIMAN

Na tabela 4.3. as linhas numeradas de 1 a 10, apresentam os resultados dos experimentos (simulações) realizadas. Cada experimento simulou o equivalente a 7 dias com turnos de 8 horas cada, tendo a duração de 3360 min.

Duas variáveis de controle foram adotadas:

1. O número de unidades produzidas no sistema, individualmente, isto é, por tipo de peça e agrupadas
2. Os tempos médios de fluxo para os cinco diferentes tipos de peças e para o total de peças produzidas.

Esta última variável foi a principal medida de desempenho adotada por Carrie.

Observando-se a tabela 4.3., verifica-se, por exemplo, que no teste ou experimento número 1, foram produzidas um total de 120 peças, divididas em 34, 25, 19, 20 e 22 unidades para as peças tipo 1, 2, 3, 4 e 5 respectivamente. É possível também observar que os tempos médios de fluxo (lead-time) das peças 1, 2, 3, 4 e 5 foram 239, 291, 202, 182 e 309 min., respectivamente, com um tempo médio para as 120 unidades de 247 min.

Ao final da tabela 4.3, na linha rotulada como Médias, apresentamos os valores correspondentes as médias dos dez experimentos.

Na tabela 4.4 pode-se observar os resultados obtidos por Carrie para as mesmas variáveis, com seu modelo programado em ECSL.

Produção p/Tipo (unid)						Tempo Médio Fluxo p/Tipo (min)					
1	2	3	4	5	Total	1	2	3	4	5	Total
30	29	16	19	25	119	242	319	211	189	328	266

Tabela 4.4: Resultados do experimento com o modelo em ECSL [CARRIE, 1988]

Aparentemente, as diferenças existentes entre os resultados obtidos, com o modelo programado em SIMAN, e aqueles obtidos por Carrie usando a linguagem ECSL são semelhantes. No item que segue, faremos um teste de hipóteses para a comparação entre as médias obtidas pelos dois modelos.

4.2.3 Teste de Hipóteses para a Comparação das Médias Obtidas

Para comparar os resultados dos valores médios obtidos pelo modelo em SIMAN e aquele desenvolvido por Carrie em ECSL, realizamos um teste t de comparação das médias com σ desconhecido.

As médias comparadas foram para os valores de produção total e tempo médio de fluxo total.

Para ambos os testes, as hipóteses H_0 e H_1 são:

- H_0 : Os valores médios obtidos pelo modelo em SIMAN e em ECSL não são diferentes;
- H_1 : Os valores médios obtidos são diferentes.

A tabela 4.5 resume os resultados dos testes:

Teste Realizado	Valor de t Calculado	Conclusão para $\alpha = 0.05$
Tempo Médio de Fluxo	0,61	Aceita-se H_0
Produção Média de Peças	-2,10	Aceita-se H_0

Tabela 4.5: Resultados dos testes de hipóteses para as médias

Os resultados dos testes de hipóteses permite-nos afirmar que as respostas apresentadas pelos dois modelos, para a solução do problema proposto, utilizando-se os mesmos parâmetros de entrada, são estatisticamente semelhantes. Podemos também afirmar, com base nestes resultados, que os dois modelos são estatisticamente equivalentes, sendo, portanto válido o modelo do SFM proposto e programado em SIMAN.

4.3 Aplicação do Modelo Proposto a Problemas de Planejamento e Controle Operacional do SFM

Uma vez validada a modelagem do SFM descrito no item 4.1, passamos agora à aplicação da metodologia proposta, isto é, do modelo de simulação incorporado do programa de alocação dinâmica de máquinas, em problemas envolvendo o planejamento e controle da produção neste sistema. Passaremos a denominar este modelo de Modelo de Planejamento Flexível (MPF).

A aplicação do MPF será feita sobre dois diferentes conjuntos de problemas exemplos, com diferentes propósitos. O primeiro conjunto de problemas, obtido junto ao trabalho de MAHESHWARI [1992], objetiva reproduzir seus experimentos e comparar seus resultados com os nossos, afim de testar o desempenho do MPF, em relação a abordagem por ele proposta. Com relação ao segundo conjunto de problemas, o propósito é explorar e demonstrar a sensibilidade do MPF, quando diferentes combinações de parâmetros operacionais são aplicados. Esta sensibilidade será posteriormente explorada, quando da

implementação da base de conhecimentos incorporada ao protótipo do sistema inteligente de simulação desenvolvido no capítulo 5.

Ambos os conjuntos de problemas exemplos foram gerados numericamente. O primeiro por Maheshwari, e o segundo por nós, usando uma adaptação de seu método.

Antes de passarmos à aplicação do modelo, apresentamos uma descrição do método empregado para a geração dos problemas exemplos, adaptado de MAHESHWARI [1992].

4.3.1 Geração Numérica de Problemas Exemplos

A necessidade de testarmos o modelo sob os mais diversos aspectos, determinou a geração numérica de problemas exemplos. Evidentemente que a validade das conclusões, baseadas nos exemplos assim gerados, dependem da qualidade destes testes. Maheshwari, em seu trabalho pesquisa, desenvolveu um gerador numérico de problemas aplicados ao planejamento e controle operacional de SFM, no qual nos baseamos, com algumas modificações, especialmente nos dados relativos ao planejamento, para gerar diversos testes.

O processo de geração de um problema é dividido em duas etapas.

Na primeira etapa, um conjunto de dados básicos para os diversos problemas são obtidos. Esta fase está subdividida em dois processos paralelos: (a) a geração de dados das peças que podem ser fabricadas e (b) a geração dos parâmetros relativos ao sistema produtivo.

Nos procedimentos gerados em (a), os dados incluem informações completas sobre os processos de fabricação associados as peças selecionadas, isto é:

- quantas e quais operações devem ser realizadas em cada uma delas
- a ordem destas operações
- respectivos tempos

O número de operações é obtido por sorteio. Os valores são discretos e uniformemente distribuídos entre 3 e 10. Os tipos de operações são designados entre os 10 tipos possíveis, também por sorteio. Os dados básicos sobre as peças completam-se pela

atribuição de tempos de processamento a cada uma das operações. Os valores são obtidos de uma distribuição uniforme discreta, com valor mínimo de 2 e máximo de 20 min.

Nos procedimentos gerados em (b), ainda na primeira etapa, o gerador define os parâmetros relativos às máquinas. São determinadas:

- as operações que cada uma delas pode realizar
- os índices de eficiência para cada par máquina:operação.

Na definição dos conjuntos Operação:Máquina, optamos pela possibilidade de cada operação ser executada em duas máquinas. Numa delas (principal) a eficiência será de 100%. Na máquina alternativa esta eficiência será de 75%. A atribuição dos índices de eficiência é também definido por sorteio entre as duas máquinas escolhidas para cada operação.

Com base nos dados gerados na primeira etapa, considerados comuns aos problemas exemplos, a geração de um novo problema inicia-se pela execução da segunda etapa. Nesta etapa três passos são executados.

- Definição do número de peças que serão manufaturadas durante um ciclo produtivo. A definição é feita por meio de um sorteio para valores discretos e uniformemente distribuídos entre 3 e 8.
- Seleção do mix entre os vinte diferentes tipos de peças possíveis de serem fabricadas. A seleção do mix é feita por meio de novo sorteio
- O último passo trata da determinação da quantidade total de peças a serem produzidas, e da participação relativa de cada um dos tipos selecionados.

Para melhor visualizar todos os procedimentos, a figura 4.2 apresenta, de forma esquemática, as duas etapas do processo de geração de problemas. As tabelas 4.6, 4.7 e 4.8 apresentam um resultado numérico da aplicação dos procedimentos.

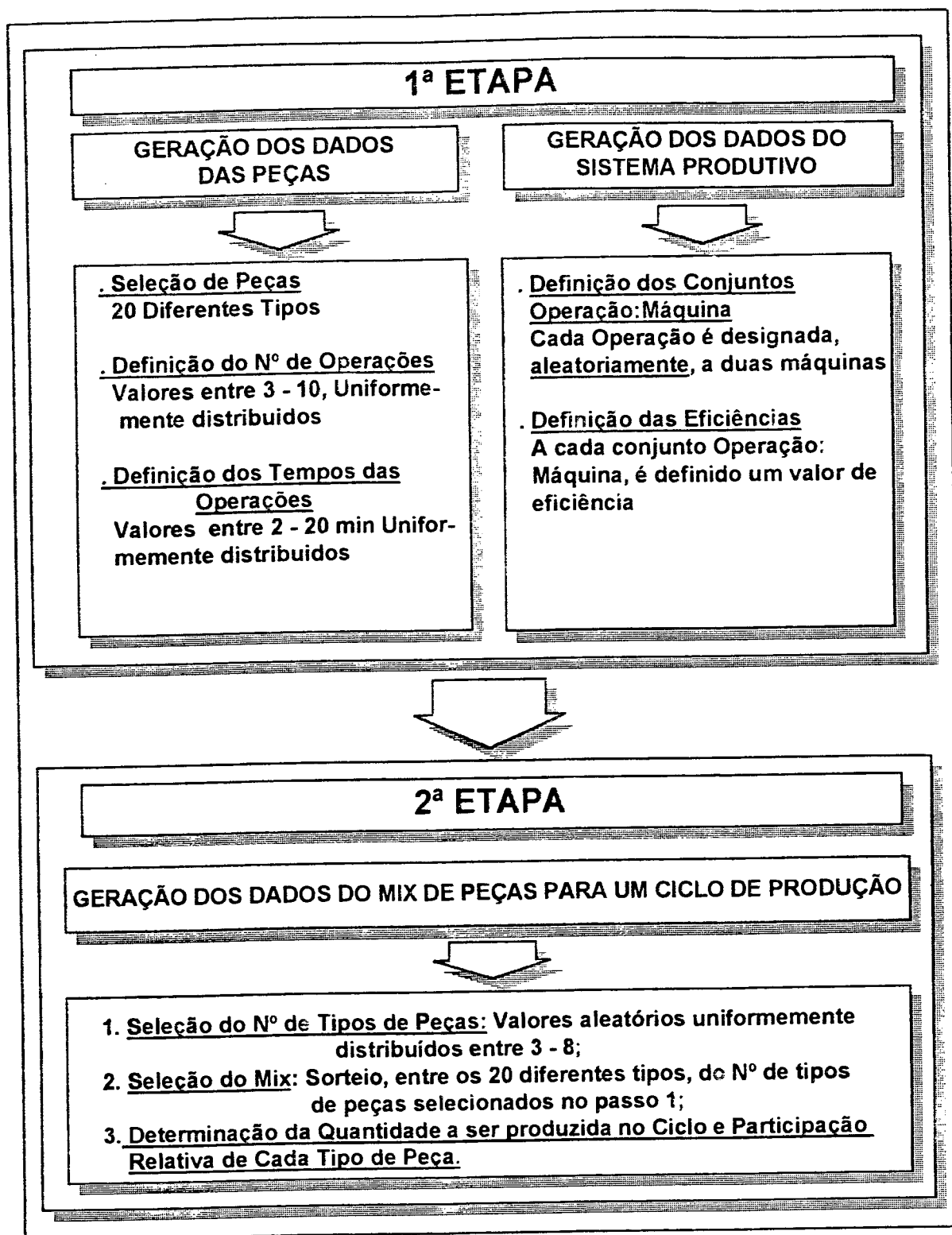


Fig. 4.2: Visão esquemática do processo de geração de problemas exemplos

A tabela 4.6 apresenta as informações relativas as peças, geradas durante a execução da primeira etapa do processo. Para os 20 diferentes tipos de peças passíveis de fabricação, observa-se na tabela: o número de operações, os tipos de operações e o tempo de cada uma delas. A peça Tipo 1, por exemplo, necessita de 7 operações para ser manufaturada. Operações 5, 10, 9, 4, 8, 6 e 7, nesta ordem. Os tempos de processamento de cada uma delas são: 3, 9, 7, 11, 12, 12 e 11 minutos respectivamente. As peças tipo 8 e 11, são manufaturadas com apenas 3 operações, enquanto que as de tipo 10 e 20 requerem 8 operações.

TIPO de PEÇA	OPERAÇÕES/TEMPOS (min)							
	1°	2°	3°	4°	5°	6°	7°	8°
1	5/3	10/9	9/7	4/11	8/12	6/12	7/11	
2	5/6	4/14	9/11	2/7	4/11			
3	3/18	2/13	4/7	1/3	5/13	4/5		
4	4/6	5/11	6/15	7/8	10/4			
5	10/13	7/4	10/12	3/3	9/11	3/18		
6	10/7	8/18	6/14	2/4	3/6	6/16		
7	1/8	2/14	1/21	7/12	10/7			
8	7/7	1/3	6/8					
9	8/14	9/11	7/17	4/20				
10	2/16	3/9	2/3	6/13	7/9	2/15	9/3	7/16
11	10/4	7/4	8/8					
12	1/10	3/6	1/11	4/12				
13	9/3	6/3	8/15	6/5	8/15			
14	4/13	7/6	10/16	3/16				
15	1/5	5/9	10/9	1/15	7/16	9/12	8/14	
16	9/4	2/6	5/15	10/11	9/11	2/8	10/17	
17	2/18	3/4	5/4	9/6	5/8	1/12		
18	2/6	6/13	8/8	9/7	5/12	6/16		
19	1/16	10/20	5/14	10/17				
20	2/15	7/8	10/5	1/16	10/12	4/7	8/11	10/12

Tabela 4.6: Operações e tempos necessários aos vários tipos de peças produzidas

A tabela 4.7 apresenta informações das eficiências atribuídas a cada uma das máquinas do SFM, quando da execução de cada uma das 10 operações. Tais informações também são geradas durante a primeira etapa do processo de criação de problemas exemplos. Observa-se, por exemplo, que a máquina 1 pode executar as operações de número 2, 4, 5, 6, 8 e 10. Para as operações de números 6, 8 e 10, esta máquina opera com máxima eficiência. Para as demais, é considerada uma máquina alternativa.

Também verifica-se que cada operação pode ser executada em duas máquinas, uma principal e outra alternativa. As máquinas alternativas possuem uma eficiência 25% menor que àquelas consideradas principais na execução de cada operação.

MÁQUINAS	EFICIÊNCIAS das OPERAÇÕES									
	1	2	3	4	5	6	7	8	9	10
1	0.00	0.75	0.00	0.75	0.75	1.00	0.00	1.00	0.00	1.00
2	0.00	1.00	0.75	0.00	0.00	0.75	1.00	0.75	1.00	0.00
3	0.75	0.00	1.00	1.00	0.00	0.00	0.75	0.00	0.00	0.75
4	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.75	0.00

Tabela 4.7: Eficiências das máquinas do SFM na realização das operações

A tabela 4.8 apresenta os dados relativos a dez problemas exemplos obtidos pela execução da segunda etapa do processo gerador.

Para cada um deles, foram sorteados valores entre 3 e 8 para a determinação do número de diferentes peças que comporão o mix. O problema 1 requer do SFM a fabricação de um mix com as peças do tipo 5, 14 e 18. Já o problema 2, requer a fabricação de um lote contendo 8 diferentes tipos de peças no seu mix: 1, 4, 6, 8, 9, 11, 15 e 18 respectivamente.

Ainda nesta etapa, são definidos o tamanho total do lote de peças a ser manufaturado, bem como, a participação relativa de cada tipo de peça do mix selecionado.

Problema Nº	Nº de Tipos de Peças	TIPOS DE PEÇAS							
		5	14	18					
1	3								
2	8	1	4	6	8	9	11	15	18
3	4	1	3	19	20				
4	4	1	6	8	14				
5	4	2	6	7	16				
6	3	3	4	12					
7	6	1	2	4	10	13	16		
8	3	9	15	16					
9	7	1	3	7	8	9	15	17	
10*	5	2	4	13	17	18			

Tabela 4.8: Um conjunto de problemas criados pela execução da segunda etapa do processo de geração de problemas exemplos

* Parte dos dados deste problema foi obtido do trabalho de Maheshwari [1992] e servirá como base para um trabalho de comparação entre os modelos e abordagens.

Uma característica importante dos dados descritos nas tabelas acima é que diferentemente daqueles gerados originalmente por MAHESHWARI [1992], aqui não são estabelecidas ligações entre peça/operações/máquinas, da maneira tradicional. O modelo de Maheshwari identifica, para cada par peça/operação, qual máquina, ou máquinas, considerando-se as alternativas, deve executar a operação. Este processo de programação das máquinas, realizado na fase de planejamento é feito a partir de um modelo de programação linear.

As diferenças entre as duas formas de abordar o problema serão melhor estabelecidas quando da apresentação dos resultados, obtidos pelo MPF, quando comparados com aqueles obtidos no trabalho de Maheshwari.

4.3.2 Comparação de Resultados

Como colocamos anteriormente, em sua pesquisa sobre o impacto das políticas operacionais no planejamento da produção em SFM, MAHESHWARI [1992] realizou um extenso trabalho, envolvendo inúmeras variáveis de controle, para concluir quais destas e,

em que nível, tinham mais importância no tratamento dos problemas daí provenientes (ver itens 3.1 e 3.2 do capítulo III).

Para a realização daquele trabalho, ele utilizou uma abordagem considerada tradicional, no sentido em que os problemas das fases de planejamento e operação são resolvidos em separado e em momentos distintos. Na fase de planejamento, empregou modelos de PL e na fase de testes operacionais, um modelo de simulação.

O SFM sobre o qual os testes foram realizados, foi o mesmo por nós modelado e validado no item 4.2 deste capítulo, tendo sido programado também na linguagem SIMAN. Este fato nos possibilita, a menos das diferenças de abordagens para a alocação dos recursos (programação linear x alocação dinâmica em tempo real), a comparação direta dos resultados quando os mesmos parâmetros operacionais são utilizados.

Para que sejam possíveis as comparações entre as duas abordagens, é preciso, no entanto, adaptar os exemplos por ele gerados para que possam ser tratados pela nova abordagem. Suas diferenças fundamentais ocorrem principalmente, no que se refere aos parâmetros de entrada dos problemas exemplos.

A abordagem usada por Maheshwari, divide-se em duas fases distintas: planejamento e operação. Na fase de planejamento, são definidos os tipos de peças que serão produzidas e em que quantidades, além de todo o sequenciamento, com a designação prévia de quais máquinas devem realizar as operações necessárias em cada peça a ser processada em um determinado ciclo. Estas designações são efetuadas por meio de modelos de programação linear, com diferentes funções objetivos (minimização de tempos, balanceamento das cargas de máquinas, etc.). Na fase de operação, o planejamento elaborado anteriormente, é simulado e testado diante de diferentes políticas operacionais.

Na abordagem por nós utilizada, a fase de planejamento se resume a determinação de que peças serão produzidas e em quais quantidades. O sequenciamento, isto é, o planejamento das operações, com conseqüente alocação das máquinas, como vimos no capítulo anterior, é executado durante a própria fase operacional, sendo determinado, com base no estado e nas restrições do sistema.

A figura 4.3 apresenta de forma resumida as diferenças entre as duas abordagens.

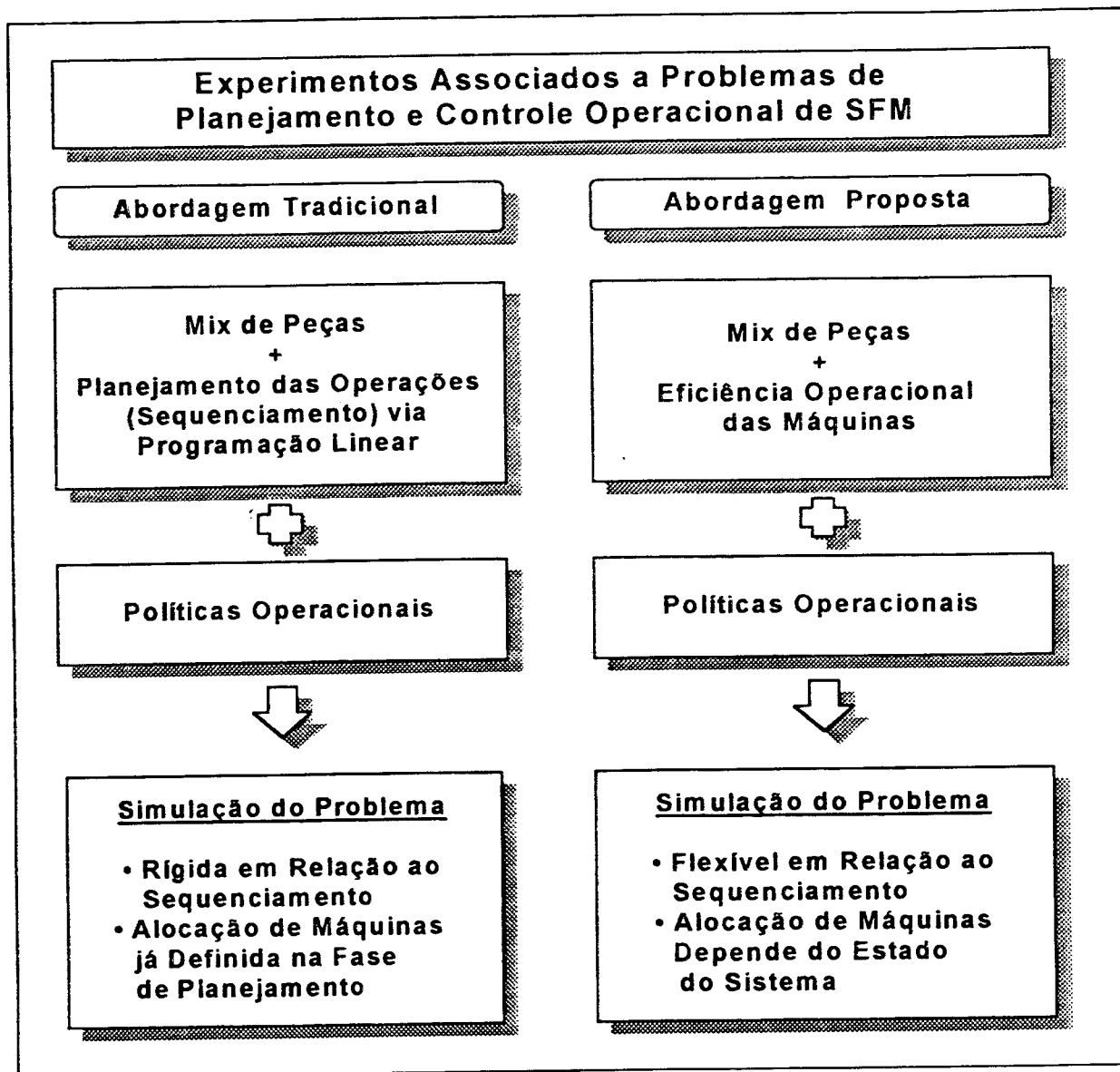


Fig. 4.3: Representação Esquemática das duas Abordagens

4.3.2.1 Experimentos Realizados

Com o objetivo de comparação dos resultados de nossos experimentos com aqueles realizados por MAHESHWARI [1992], apresentamos nesta seção um exemplo completo de aplicação das duas abordagens. O motivo da escolha deste exemplo é ter sido este o único problema exemplo, apresentado pelo autor, documentado em toda as suas fases.

4.3.2.2 Problema Exemplo Utilizado na Comparação de Resultados

O problema exemplo a ser aqui apresentado é o de número 10 da tabela 4.8. Ele estabelece a fabricação de um lote com um mix constituído pelas peças de número 2, 4, 13, 17 e 18. As operações necessárias a sua manufatura e os tempos destas, podem ser observados na tabela 4.6. A quantidade total a ser produzida é de 108 peças. A participação relativa de cada um dos tipos deve ser de 22%, 17%, 11%, 22% e 28% respectivamente. Os dados sobre o mix e participação relativa foram gerados pela 2ª etapa do processo de geração de problemas exemplos.

Para simularmos o ciclo de produção deste lote, usando nossa abordagem e modelo, necessitamos ainda dos elementos da tabela 4.7, eficiências das máquinas, juntamente com uma definição das políticas operacionais para o despacho de peças e veículos além dos números relativos aos buffers e pallets.

Já a simulação deste problema usando a abordagem de MAHESHWARI [1992], necessita a geração prévia do planejamento da carga das máquinas.

Uma vez que nosso modelo procura equilibrar, durante a simulação, o balanceamento da carga das máquinas, a comparação entre resultados operacionais será feita com aqueles obtidos por MAHESHWARI [1992], nos quais ele aplica o mesmo tipo de função objetivo ao modelo de PL para a geração do planejamento da carga das máquinas, o qual também considera a possibilidade de máquinas alternativas.

4.3.2.3 Medidas de Desempenho Adotadas na Comparação dos Resultados

Conforme visto no item 3.3 do capítulo anterior, são várias as medidas de desempenho possíveis de serem adotadas para a avaliação operacional dos SFM. De acordo com a literatura [LEUNG, 1990; GUPTA, 1992], as duas principais medidas são o tempo médio de passagem ou de fluxo das peças pelo sistema e o tempo total do ciclo produtivo, isto é, o tempo total necessário ao sistema para produzir o lote de peças. LEUNG [1990] mostra em seu trabalho que adotar o tempo total do ciclo produtivo como medida de desempenho em SFM que produzem um número finito de peças em um dado ciclo, é equivalente a medir-se a utilização total do sistema ou sua taxa de produção. Segundo ele, a minimização do tempo do ciclo de produção equivale à maximização da taxa de utilização

do sistema. Com base nestes argumentos, estas foram as duas principais medidas de desempenho adotadas por MAHESHWARI [1992] e que faremos uso na comparação dos resultados.

4.3.2.4 Resultados Obtidos nas Comparações

A tabela 4.9 apresenta resultados comparativos do desempenho operacional do SFM, para várias combinações de políticas operacionais, tamanho de buffers e número de pallets, para o problema exemplo, quando da aplicação das duas abordagens.

Experimento	RLP	RDV	VAG	PAL	BUF	TCP			TFluxo		
						AT	NA	$\Delta\%$	AT	NA	$\Delta\%$
1	1	1	2	12	6	2429	2172	10,6	n/d	145	
2	3	4	2	8	5	2368	2109	10,9	157	148	5,7
3	3	2	2	10	4	2290	1989	13,1	180	164	8,9
4	3	4	2	12	4	2659	2214	16,7	152	131	13,8
5	3	2	2	6	5	2549	2402	5,8	130	121	6,9
6	4	4	2	10	6	2578	2163	16,1	210	195	7,1
7	4	2	2	8	3	2922	2562	12,3	194	180	7,2
8	4	4	2	6	3	3013	2630	12,7	197	185	6,1
9	4	2	2	12	6	2681	2108	21,4	142	138	2,8

Tabela 4.9: Resultados dos experimentos quando se aplicam as duas abordagens

RLP: Regras p/Liberação de Peças*

RDV: Regras p/Despacho de Veículos*

VAG: N° de VAGs

PAL: N° de Pallets

BUF: Espaço nos Buffers

TCP: Tempo do Ciclo de Produção

TFluxo: Tempo Médio de Fluxo

AT: Abordagem Tradicional proposta por MAHESHWARI [1992]

NA: Nova Abordagem proposta neste trabalho

$\Delta\%$: Redução nos tempos de ciclo e de fluxo, em termos percentuais, obtida pelo uso de NA em relação a AT

* Para detalhes sobre a legenda ver figura 3.2 do capítulo III, Elementos do controle operacional de SFM.

1: MTxP, 2: MNV, 3: MTTP, 4: OA

1: MCTFD, 2: MNVR 3: MNPS, 4: FIFO

Nos experimentos originalmente apresentados por MAHESHWARI [1992], as várias combinações entre as variáveis operacionais seguem um projeto de experimentos gerados

pelo método de TAGUCHI [1987]. No entanto, nem todos os experimentos apresentados nos interessam, uma vez que muitas das experimentações por ele realizadas tornam-se irrelevantes para fins de comparação, por força do modelo por nós idealizado. Desta forma, a tabela mostra somente aqueles experimentos cuja função objetivo utilizada (Balanceamento da Carga de Máquinas), regras de liberação das peças e regras de despacho de veículos, possuem equivalentes em nosso modelo. Os demais experimentos por nós realizados são apresentados na seção 4.3.3 (Análise de Sensibilidade do Modelo). A propósito, tanto os resultados devidos a MAHESHWARI [1992], quanto aqueles resultantes da aplicação do MPF, são valores médios de simulações, com pelo menos 10 repetições do mesmo experimento, mas com diferentes sementes associadas aos algoritmos geradores de números aleatórios.

4.3.2.5 Observações sobre os Resultados Obtidos

Na tabela 4.9 podemos observar que, tanto os valores associados aos tempos do ciclo produtivo (TCP) quanto aqueles associados aos tempos médios de fluxo (TFluxo), são menores quando se emprega a abordagem proposta neste trabalho.

As reduções nos tempos de ciclo e de fluxo, em termos percentuais, encontram-se sob as colunas referidas com o símbolo $\Delta\%$. Embora estas reduções tenham ocorrido em 100% dos exemplos tratados, os ganhos obtidos não são homogêneos, tendo variado de 5,8% a 21,4% nos tempos dos ciclos de produção e de 2,8% a 13,8% nos tempos médios de fluxo, dependendo dos parâmetros operacionais aplicados.

Observamos que as diferenças a favor da nova abordagem são mais acentuadas quando o sistema se apresenta mais congestionado, isto é, quando são maiores os tamanhos dos buffers e o número de pallets no sistema. A causa destes ganhos pode estar relacionada com a falta de flexibilidade do modelo adotado por MAHESHWARI [1992], uma vez que nele a decisão relativa à alocação de máquinas é feita “a priori”, via programação linear, desconsiderando a dinâmica operacional do sistema, a qual tem forte dependência dos valores atribuídos aos recursos secundários do sistema.

Por outro lado, por realizar dinamicamente a alocação dos recursos, o modelo proposto é mais sensível à potencial flexibilidade dos SFM, especialmente à flexibilidade de rotas e,

portanto, mais adaptável aos eventuais congestionamentos causados pelo maior número de peças no sistema.

Nos casos em que maiores restrições ao número de peças no sistema foram impostos, com menores chances de congestionamentos, os ganhos nos tempos de ciclo de produção foram menores uma vez que, tanto a programação linear quanto o modelo dinâmico, chegam a resultados semelhantes para a alocação de máquinas.

O gráfico 4.1 apresenta os dados relativos aos ganhos obtidos. Observa-se a não existência de paralelos entre os ganhos obtidos nos tempos do ciclo e nos tempos de fluxo. Tais tempos dependem, sobretudo dos valores designados aos buffers e pallets e, muitas vezes são objetivos antagônicos, como será mostrado na seção 4.3.3.

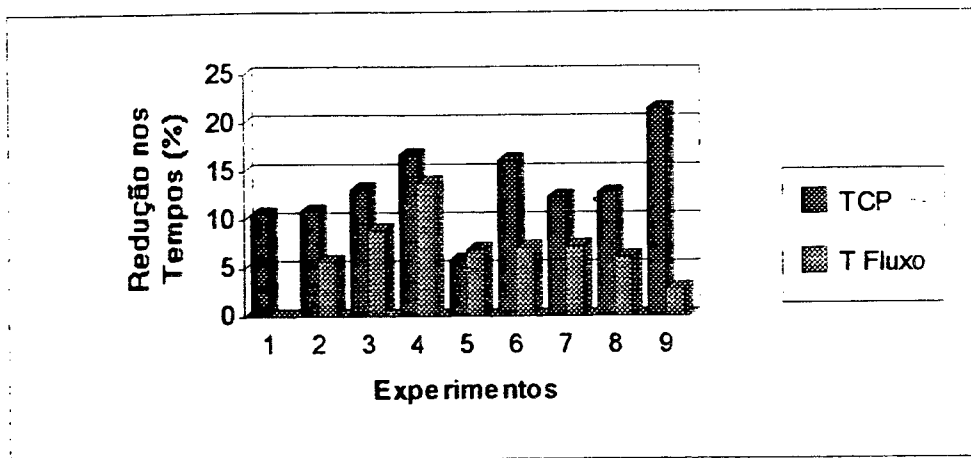


Gráfico 4.1: Ganhos Percentuais Obtidos pelo uso do Novo Modelo

4.3.2.6 Testes de Hipóteses para Comparação das Médias obtidas nos Experimentos

Com o objetivo de validar estatisticamente os resultados obtidos nos experimentos apresentados na tabela 4.9, realizamos testes de hipóteses para comparação das médias das duas medidas de desempenho, quando da aplicação das duas abordagens.

No primeiro teste, as hipóteses H_0 e H_1 referem-se as médias dos tempos de ciclo de produção.

H₀: Os tempos médios do ciclo de produção obtidos nas abordagens AT e NA são iguais.

H₁: O tempo médio do ciclo de produção obtido na abordagem AT é maior que tempo médio do ciclo de produção obtido na abordagem NA.

Calculando-se o valor de t para as diferenças entre os valores de TCP referentes a AT e NA obtemos: $t = 8,38$.

Repetimos o mesmo teste para os tempos médios de fluxo obtidos pelas duas abordagens. As duas hipóteses são semelhantes, a menos da medida de desempenho avaliada.

H₀: Os tempos médios de fluxo das peças obtidos nas abordagens AT e NA são iguais.

H₁: O tempo médio de fluxo das peças obtido na abordagem AT é maior que o tempo médio de fluxo das peças obtido na abordagem NA.

Calculando-se o valor de t para as diferenças entre os valores de TCP referentes a AT e NA obtemos: $t = 6,79$.

Na tabela 4.10 abaixo apresentamos um resumo dos resultados e conclusões dos testes.

Variável Analisada	Valor Calculado de t	Conclusão com $\alpha=0,05$
TCP	8,38	Rejeita-se H ₀
TFluxo	6,79	Rejeita-se H ₀

Tabela 4.10: Resultados dos testes de comparação das médias de TCP e Tfluxo

Pelos resultados obtidos para os valores de t , não é possível comprovar H₀ em nenhum dos dois testes realizados. Podemos então afirmar que, estatisticamente, os tempos obtidos pela aplicação da nova abordagem são menores que aqueles obtidos por MAHESHWARI [1992] em sua abordagem tradicional.

4.3.3 Análise de Sensibilidade do Modelo

Na seção 4.2, vimos que são promissores os resultados obtidos pelo modelo de simulação associado com o programa de alocação dinâmica de recursos. É necessário, no entanto complementarmos aqueles testes, para verificarmos sua sensibilidade com relação as diversas combinações de parâmetros e políticas operacionais, determinando quais delas resultam em um melhor desempenho do sistema.

Com a realização destes testes, será possível, também, verificarmos se o impacto das políticas operacionais sobre o desempenho do SFM, observados nos trabalhos de MAHESHWARI [1992] e GUPTA [1992], coincidem com aqueles por nós observados, quando da aplicação de nosso modelo.

Para a realização destas análises, um total de 576 simulações foram efetuadas, sobre três dos diferentes problemas exemplos gerados. Testamos as possíveis combinações ($4 \times 3 \times 4 \times 4$) entre as políticas e parâmetros operacionais, isto é as regras de liberação de peças para o sistema (4), as regras para despacho de veículos (3), tamanho dos buffers (4) e do número de pallets (4).

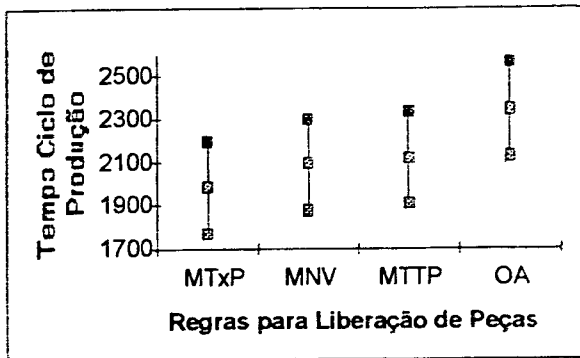
Os experimentos realizados são semelhantes àqueles efetuados por e GUPTA [1992]. MAHESHWARI [1992]. As diferenças ficam por conta de um fator extra utilizado naqueles trabalhos, as funções objetivo, os quais possuem dois e quatro diferentes níveis respectivamente. Como foi dito anteriormente, o modelo proposto objetiva primordialmente balancear as cargas das máquinas.

Algumas das combinações entre políticas operacionais testadas por eles também não foram explicitamente verificadas por estarem embutidas dentro do programa de alocação de recursos ou do próprio modelo de simulação, como por exemplo aquelas envolvendo as políticas de priorização de peças nos buffers.

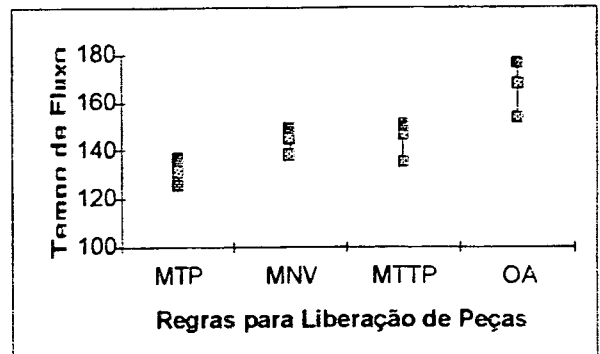
4.3.3.1 Observações sobre os Resultados das Simulações

Dois conjuntos de regras operacionais e dois recursos secundários dos SFM foram tratados e analisados, individualmente, para verificar suas influências sobre o desempenho dos SFM.

Os gráficos 4.2 a 4.5, apresentam os resultados dos valores médios dos tempos do ciclo de produção (A) e dos tempos de fluxo (B), obtidos nas simulações realizadas. Nestes, é possível verificar a influência, individual, dos principais fatores (políticas e parâmetros operacionais), sobre estas medidas de desempenho. Os três pontos apresentados em cada segmento dos gráficos, representam os valores mínimo, médio e máximo, respectivamente.

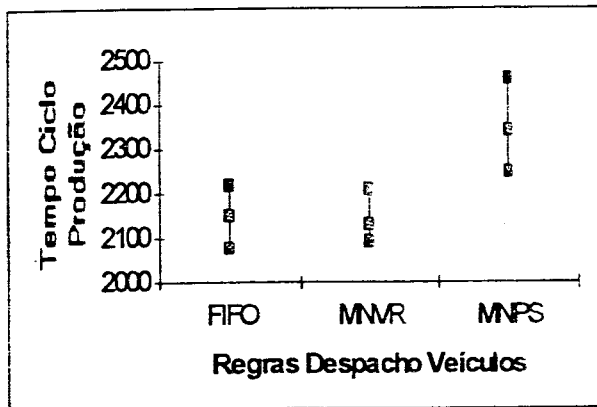


(A)

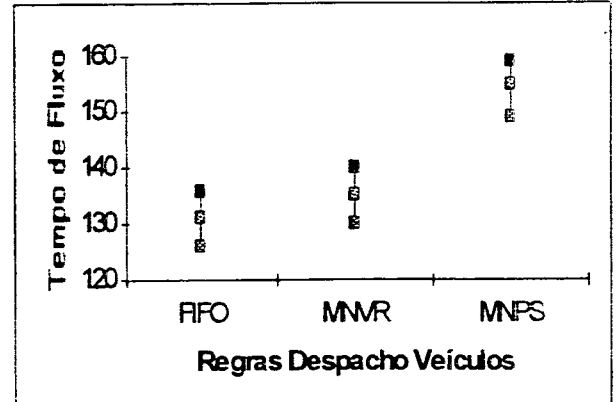


(B)

Gráfico 4.2: Efeito das Regras para a Liberação de Peças

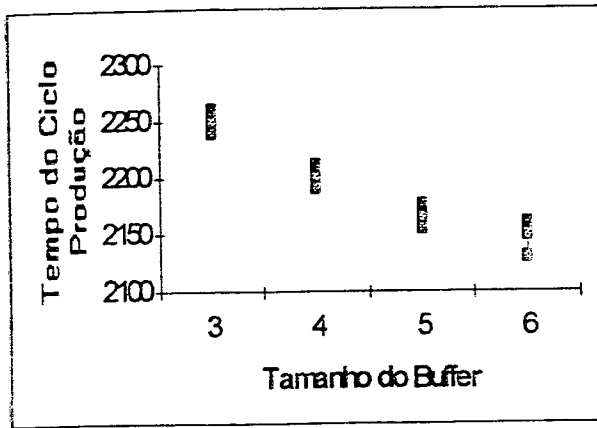


(A)

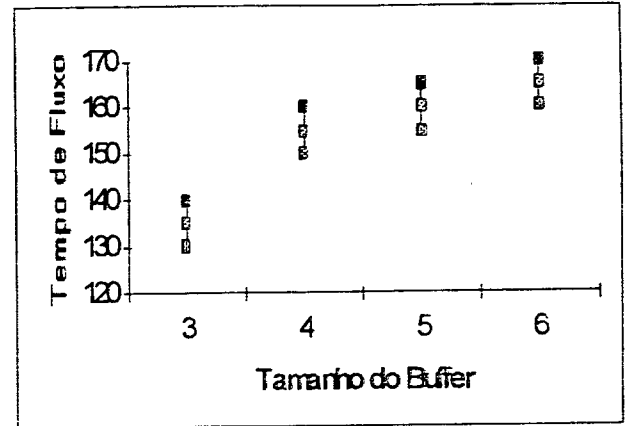


(B)

Gráfico 4.3: Efeito das Regras para Despacho de Veículos

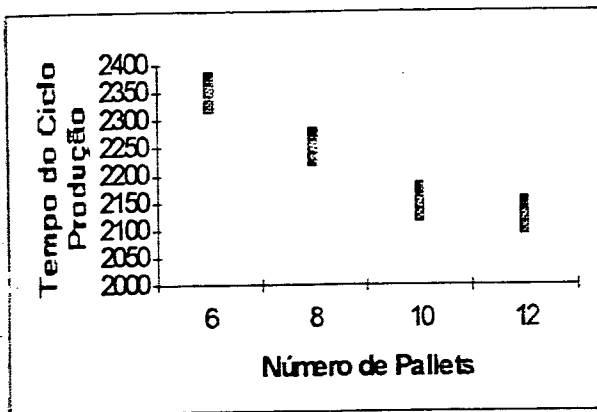


(A)

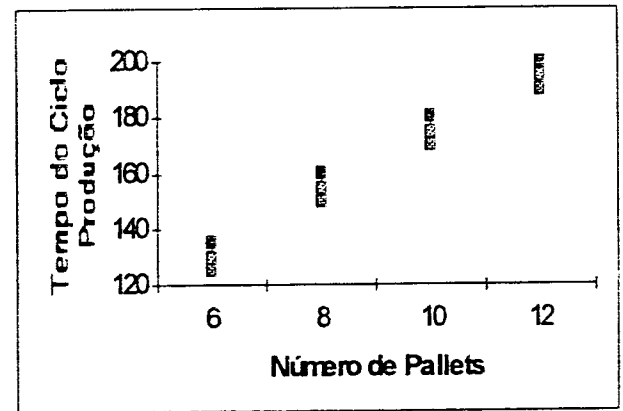


(B)

Gráfico 4.4: Efeito do Tamanho dos Buffers



(A)



(B)

Gráfico 4.5: Efeito do Número de Pallets

As observações sobre os efeitos dos quatro fatores sobre as medidas de desempenho adotadas, são comentadas abaixo.

Efeito das Regras de Liberação de Peças:

O efeito destas regras é apresentado no gráfico 4.2 (A e B). A regra que propicia os melhores resultados em ambos os casos (TCP e TFluxo), é a MTxP (Menor Taxa de Produção), pela qual as peças são introduzidas no sistema de tal forma a manter sempre em equilíbrio o mix desejado. Duas outras regras, MNV (Menor Número de Visitas) e MTTP (Menor Tempo Total de Processamento), obtiveram resultados estatisticamente semelhantes, uma vez que as duas regras mantêm uma ordem cíclica e só diferem a ordem

no início das liberações das peças ao sistema. O pior desempenho é a ordem aleatória (OA). Este desempenho da OA, serve para confirmar que o controle da liberação de peças, principalmente em SFM fechados, pode melhorar o desempenho operacional do sistema.

O melhor desempenho da regra MTxP, deve-se ao fato dela manter o balanço do mix de peças o que permite manter balanceada carga relativa dos centros de usinagem.

Efeito das Regras para Despacho de Veículos:

O efeito das regras de despacho sobre o desempenho do sistema é apresentado no gráfico 4.3 (A e B). Para estes testes, não utilizamos todas as regras listadas no item 3.1.2.3 do capítulo III (Políticas para o Despacho de Veículos Transportadores). A aplicação da regra MCTFD (Menor Carga de Trabalho na Fila de Destino) produz, na prática, efeito equivalente ao que o próprio modelo de simulação desenvolvido produz. Interessante notar que em seu trabalho, Maheshwari relata que esta foi também a regra que melhor resultados produziu sobre o desempenho do SFM. Isto reforça que os resultados obtidos pelo MPF são melhores quando comparados com aqueles por ele obtidos.

Entre as regras por nós testadas, FIFO e MNVR (Menor Número de Visitas Restantes) obtiveram resultados estatisticamente semelhantes. Ambos tiveram desempenho bem superior a regra MNPS (Máximo Número de Peças na Saída). MNPS, é uma regra que procura evitar o acúmulo de peças aguardando por transporte na máquina, e com isso evitar o bloqueio do sistema. Como o bloqueio depende substancialmente do número de pallets e tamanho dos buffers, um controle sobre estas variáveis evita a ocorrência dos bloqueios, tornando a regra desnecessária.

Impacto do Tamanho dos Buffers:

Os efeitos dos tamanhos dos buffers são apresentados no gráfico 4.4 (A e B). Sua influência é significativa sobre as duas medidas de desempenho. O aumento do tamanho dos buffers, reduz o tempo do ciclo de produção, o que é equivalente a um aumento na taxa de produção do sistema. Por outro lado, o aumento dos buffers, aumenta o tempo de fluxo das peças pelo sistema. Esta é uma decisão que depende, portanto, do tipo de objetivo do decisor. Nos sistemas do tipo *puxados*, deseja-se, geralmente, pequenos

espaços nos buffers, propiciando uma rápida passagem das peças pelo sistema. Já para os sistemas do tipo *empurrados*, a falta de espaço nos buffers pode resultar em um bloqueio do sistema, uma vez que se não houver espaço na máquina destino o deslocamento das peças não pode ocorrer.

Impacto do Número de Pallets:

O efeito do número de pallets é mostrado no gráfico 4.5 (A e B). Assim como o tamanho dos buffers, também o impacto desta variável é significativo para ambas as medidas de desempenho. O tempo do ciclo de produção se reduz com o aumento do número de pallets, enquanto que o tempo de fluxo se amplia. Estes resultados se explicam por si só. O aumento dos pallets, reduz o tempo de ociosidade das máquinas, aumentando sua taxa de utilização e, simultaneamente, o tempo em que as peças devem esperar em filas.

Resumo das Observações Realizadas:

Considerando todos os efeitos medidos com os vários fatores sobre as medidas de desempenho operacional do SFM, verificou-se que:

- MTxP é a regra de liberação de peças dominante para as duas medidas;
- Considerando as duas medidas, FIFO pode ser bem empregada como regra para o despacho de veículos, uma vez que, internamente, o modelo já realiza a busca das máquinas com menor carga de trabalho;
- O tamanho dos buffers e número de pallets vão depender dos objetivos dos decisores.

As tendências verificadas acima, são similares aquelas observadas por MAHESHWARI [1992] em seu trabalho de pesquisa. As únicas diferenças encontradas são com relação aos tempos médios de ciclo e fluxo. De maneira geral, nosso modelo resultou em números médios 14% menores para os tempos de ciclo produtivo e cerca de 6% para os tempos de fluxo, quando o sistema utilizou dois VAGs para o transporte. Maheshwari não apresentou resultados para outros números de VAGs.

4.3.4 Busca das Melhores Combinações entre os Elementos Operacionais

Uma vez que temos conhecimento de quais os principais fatores que influem no desempenho do SFM, torna-se interessante buscar aquela combinação de elementos que apresente os melhores resultados operacionais.

Com este propósito, realizamos uma série de novos experimentos, utilizando MTP como regra para a liberação de peças e FIFO para o despacho de veículos. Fixadas estas regras, o tamanho dos buffers, o número de pallets e o número de veículos transportadores, foram combinados, de todas as formas possíveis dentro de determinados limites. O mesmo sistema flexível de manufatura foi utilizado e os testes consideraram o problema descrito em 4.3.2.2. Os resultados destes experimentos foram obtidos com a realização de aproximadamente 190 diferentes combinações de fatores. Cada combinação foi simulada com dez diferentes sementes geradoras de números aleatórios. Os resultados foram tomados pela média destes valores. Abaixo apresentamos os resultados e comentamos sobre os novos experimentos realizados.

4.3.4.1 Resultados e comentários sobre os novos experimentos realizados

Os gráficos 4.6 a 4.17, apresentam os resultados dos valores médios de quatro diferentes medidas de desempenho: Tempo Médio do Ciclo de Produção (TCP), Tempo Médio de Fluxo das Peças (TMFluxo), Taxa Média de Utilização das Máquinas (TxMUtilMaq) e Taxa Média de Utilização dos VAGs (TxMUtilVAGs). Os resultados, obtidos nas simulações realizadas, mostram, a influência das combinações dos principais fatores (políticas e parâmetros operacionais), sobre o desempenho operacional do sistema.

- *Relação entre Tamanho dos Buffers, Número de Pallets e Tempo do Ciclo de Produção, quando 2, 3 e 4 VAGs são utilizados.*

Nos gráficos 4.6, 4.7 e 4.8, observamos o efeito da variação dos tamanhos dos buffers e do número de pallets sobre o tempo médio do ciclo de produção (TCP), quando são utilizados 2, 3 e 4 VAGs, respectivamente. O comportamento da medida de desempenho é semelhante nos três gráficos. Verificamos que para cada valor de tamanho de buffer, na medida em que aumentamos o número de pallets, há uma redução no tempo do ciclo de

produção. Contudo, observamos que esta redução não é contínua, existindo um ponto de inflexão na curva.

O ponto de inflexão corresponde a um estado de saturação do sistema, resultando em congestionamentos e aumento do tempo médio do ciclo de produção. De maneira geral, o aumento dos valores da combinação tamanho dos buffers x número de pallets, leva a uma redução do TCP, até o ponto de saturação. Outra observação importante é revelada quando olhamos para os três gráficos em conjunto. Nestes gráficos, a principal variável é o número de VAGs. Verificamos que na medida em que crescem os valores da combinação buffers x pallets x VAGs, obtém-se melhores resultados para o TCP. Tais resultados, porém, tem um custo associado. Como se verá em outras análises, as taxas de utilização dos veículos transportadores e dos centros de usinagem, sobem para valores inaceitáveis. Embora não tenhamos pesquisado os custos associados a implantação de um maior número de VAGs, verificou-se que, embora existam ganhos no TCP, os valores encontrados não são substanciais e, provavelmente, não justificariam um aumento crescente do número de VAGs, como pode ser visto no gráfico 4.8.

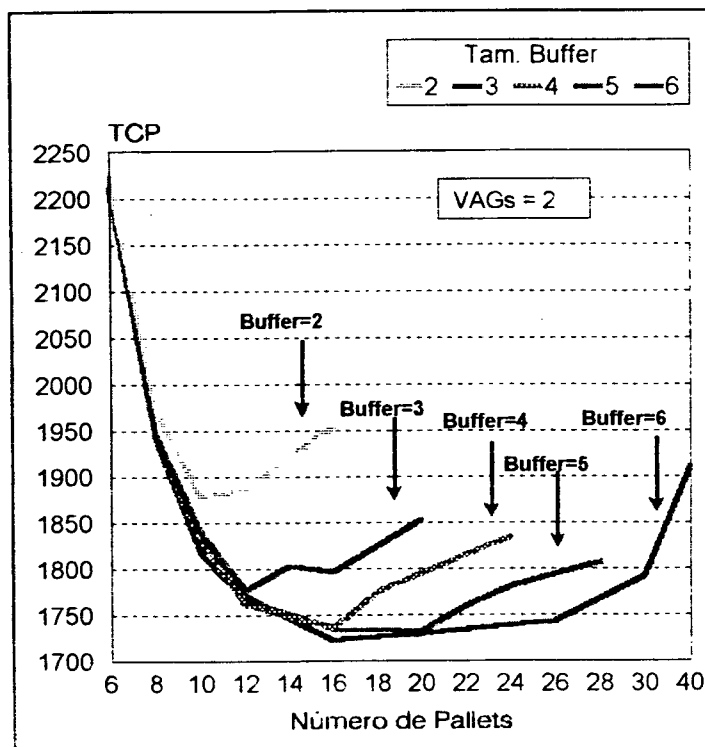


Gráfico 4.6: Resultados para o uso de 2 VAGs

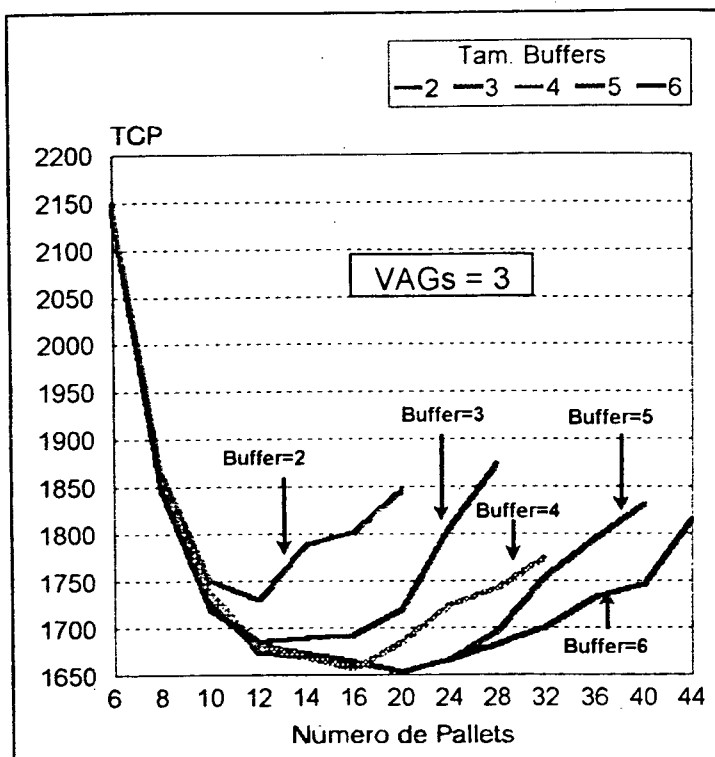


Gráfico 4.7: Resultados para o uso de 3 VAGs

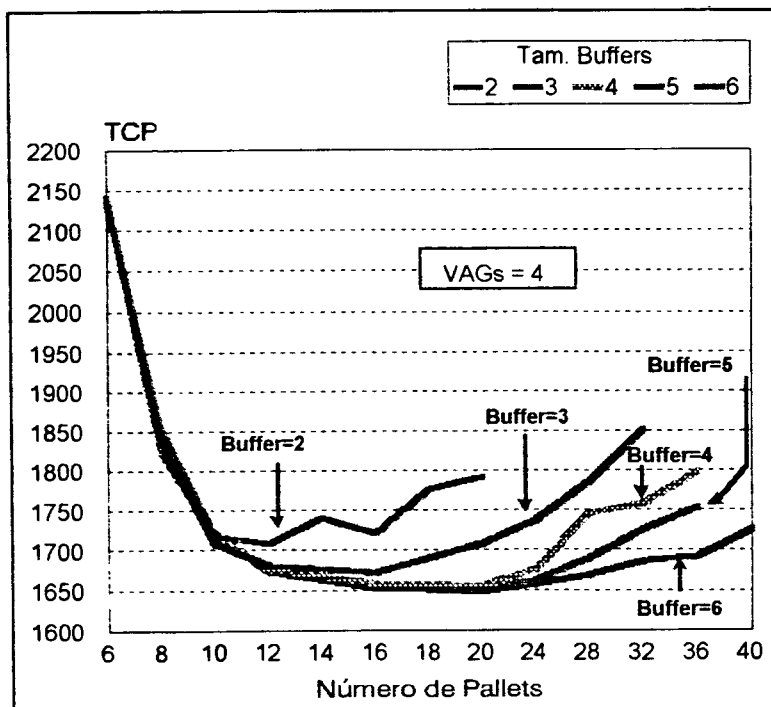
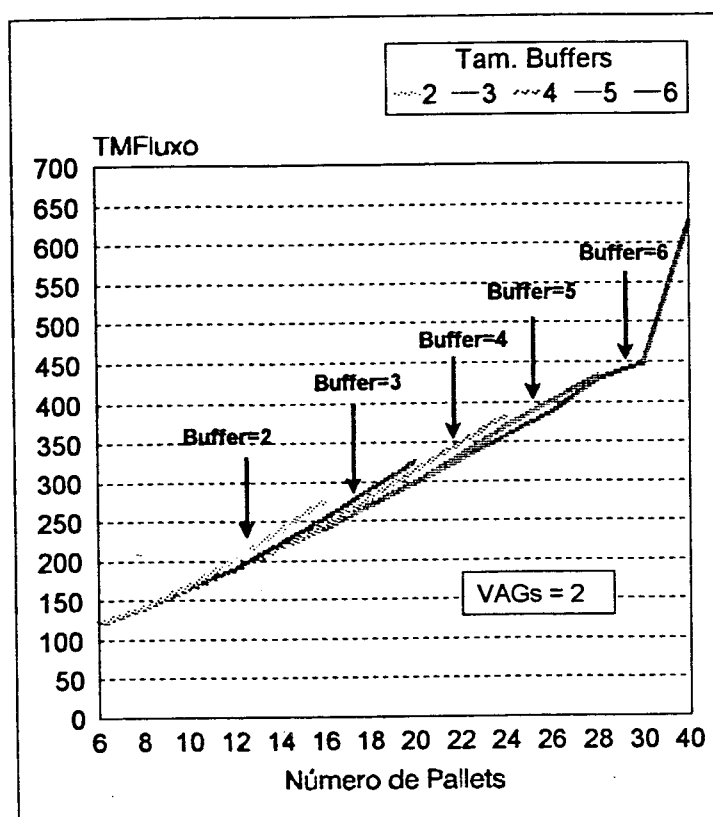


Gráfico 4.8: Resultados para o uso de 4 VAGs

- *Relação entre Tamanho dos Buffers, Número de Pallets e Tempo Médio de Fluxo quando, 2, 3 e 4 VAGs são utilizados.*

Nos gráficos 4.9, 4.10 e 4.11, observamos o efeito da variação dos tamanhos dos buffers, do número de pallets sobre o tempo médio de fluxo das peças quando são utilizados 2, 3 e 4 VAGs, respectivamente. O comportamento da medida de desempenho é semelhante nos três gráficos. Verificamos que, para cada valor de tamanho dos buffers, na medida em que aumentamos o número de pallets, há um aumento do tempo de fluxo das peças no sistema.

De maneira geral, o aumento dos valores da combinação tamanho dos buffers x número de pallets x número de VAGs, leva a um crescimento do tempo médio de fluxo das peças, embora esta sensibilidade seja pequena. Estes resultados já haviam sido verificados por MAHESHWARI [1992], por GUPTA[1992]e, por nós mesmos quando da realização dos testes comparativos e das análises sobre as influências individuais de cada um dos fatores (ver gráficos 4.4 e 4.5).



Gráficos 4.9: Resultados para 2 VAGs

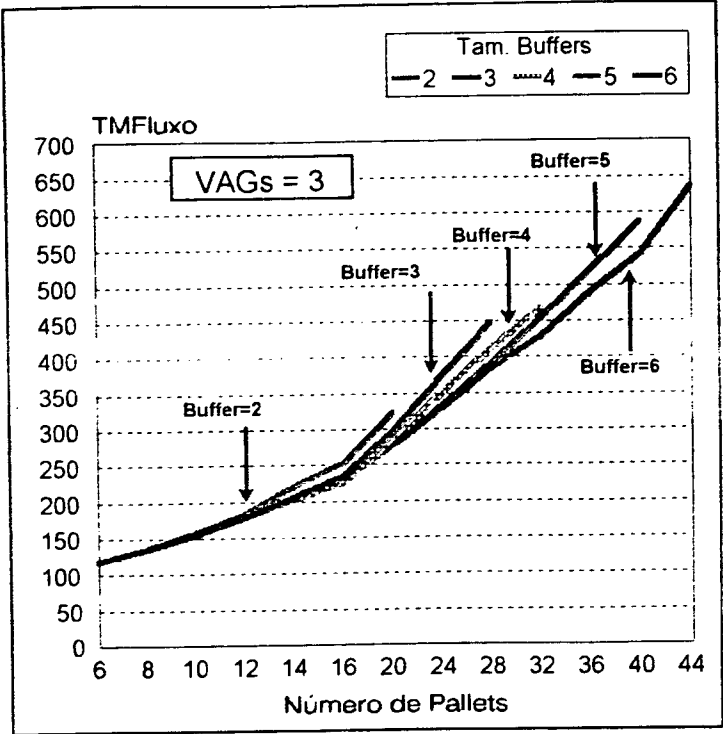


Gráfico 4.10: Resultados para 3 VAGs

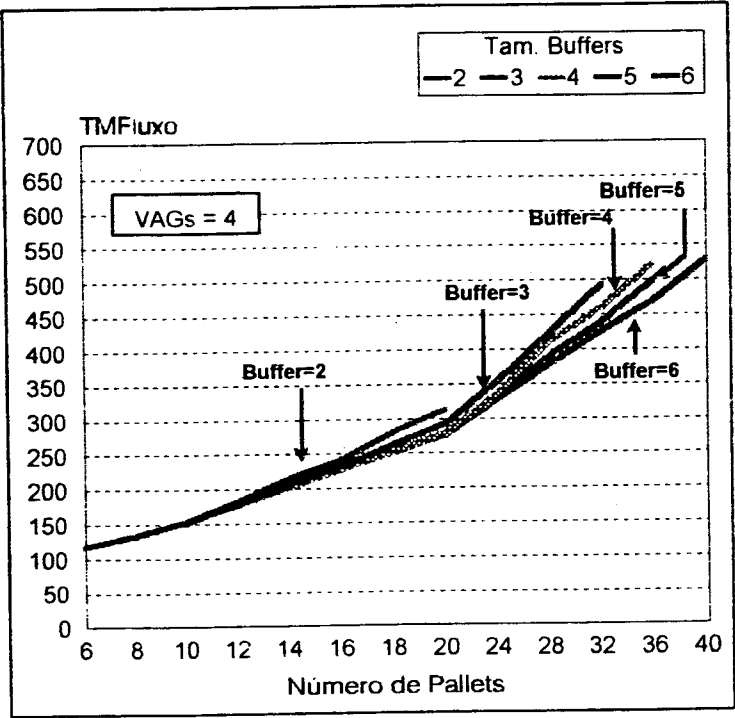


Gráfico 4.11: Resultados para 4 VAGs

- *Relação entre Tamanho dos Buffers, Número de Pallets e Taxa Média de Utilização das Máquinas, quando 2, 3 e 4 VAGs são utilizados.*

Os gráficos 4.12, 4.13 e 4.14, apresentam os resultados da medida de desempenho, Taxa Média de Utilização das Máquinas, para as diversas combinações dos tamanhos dos Buffers, número de Pallets e número de VAGs.

Como podemos observar, para cada combinação tamanho dos buffers x número de pallets, obtemos curvas com pontos de inflexão. Comparando estas curvas com àquelas obtidas nos gráficos 4.6, 4.7 e 4.8, sobre os tempos médios dos ciclos de produção (TCP), verificamos uma coincidência nos valores das duas variáveis para tais pontos.

A razão destas inflexões, é a mesma encontrada anteriormente, isto é, na medida da ocorrência de congestionamentos no sistema, um maior número de peças encontram-se nos buffers das máquinas. Uma vez que a capacidade do sistema não se altera, o tempo necessário às máquinas para a realização de todas as operações nas peças, continua o mesmo. No entanto, em função do congestionamento, o tempo do ciclo de produção cresce, e a taxa de utilização das máquinas, dada pela relação entre o tempo de uso efetivo das máquinas e o tempo total do ciclo de produção, diminui.

Podemos também observar que na medida em que aumenta o número de VAGs, cresce a taxa média de utilização das máquinas, chegando, muitas vezes a valores acima dos 95%.

Esta alta taxa média de utilização das máquinas, associada aos congestionamentos no sistema, pode ser apontada como uma das limitações à utilização de grandes números nas combinações entre número de pallets, tamanho de buffers e número de veículos transportadores.

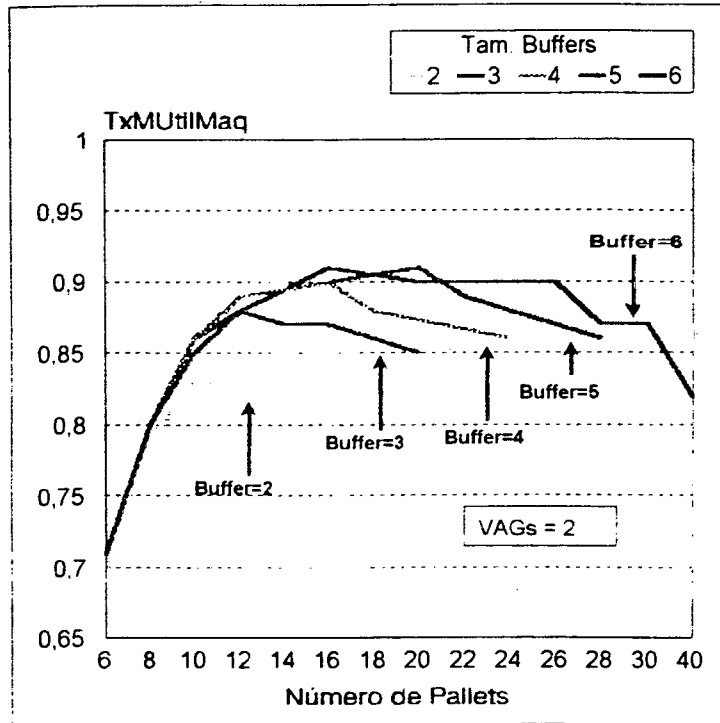


Gráfico 4.12: Resultados para 2 VAGs

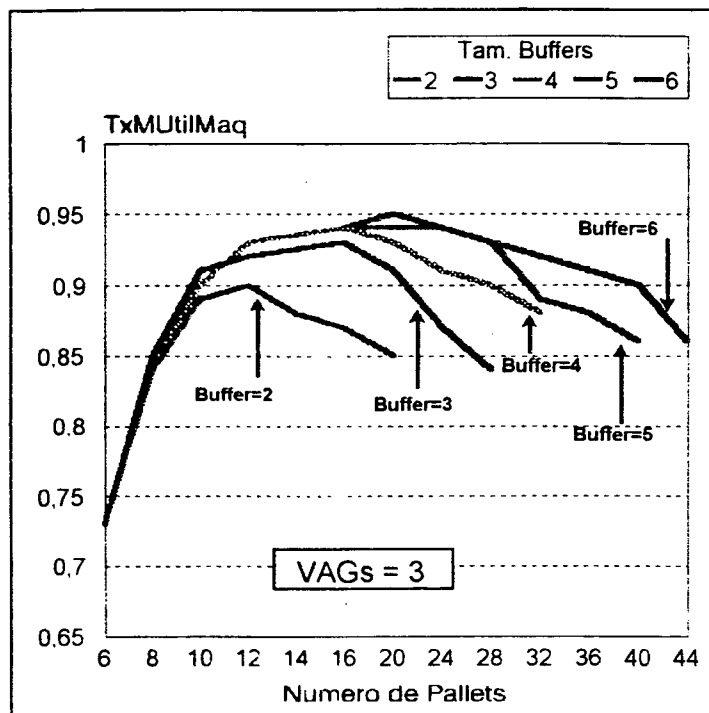


Gráfico 4.13: Resultados para 3 VAGs

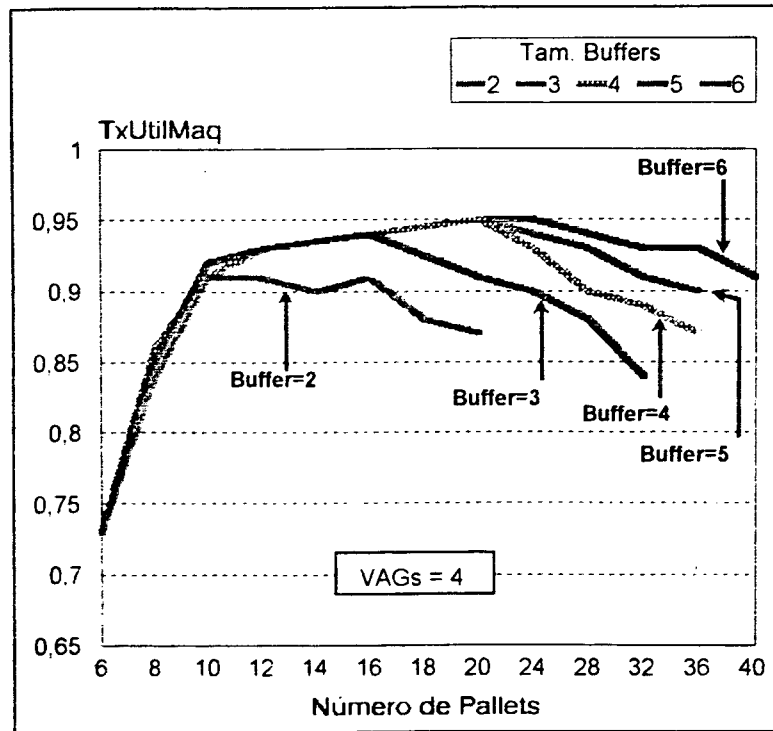


Gráfico 4.14: Resultados para 4 VAGs

- *Relação entre Tamanho dos Buffers, Número de Pallets e Taxa Média de Utilização dos VAGs, quando 2, 3 e 4 VAGs são utilizados.*

Os gráficos 4.15, 4.16 e 4.17, apresentam os resultados da medida de desempenho, Taxa Média de Utilização dos VAGs, para as diversas combinações dos tamanhos dos Buffers, número de Pallets e número de VAGs.

De maneira geral observamos em todos os gráficos uma tendência a uma alta taxa de utilização dos VAGs, na medida do crescimento do número de pallets e do tamanho dos buffers. Isto se explica porque os aumentos nestes dois fatores propiciam a entrada de mais peças no sistema para serem transportadas. Notamos, no entanto, que a forma das curvas não são as mesmas para os três gráficos. No primeiro, 4.15, que apresenta os resultados para 2 VAGs, as taxas de utilização dos VAGs crescem aos saltos quando aumentamos o número de pallets de 6 para 8 e de 8 para 10. Os demais valores já não

apresentam diferenças de taxas tão expressivas. O motivo destes resultados, é que com apenas 2 veículos e um crescimento grande do número de peças no sistema devido ao número de pallets e ao tamanho dos buffers, é eliminada a ociosidade do sistema de transporte, com as taxas de utilização chegando perto dos 100%.

Nos gráficos 4.16 e 4.17, percebemos que o salto para a saturação do sistema de transportes ocorre em pontos além dos 10 ou 12 pallets no sistema. No caso de 4 VAGs, com tamanhos de buffers iguais a 6, o salto só ocorre acima dos 16 pallets.

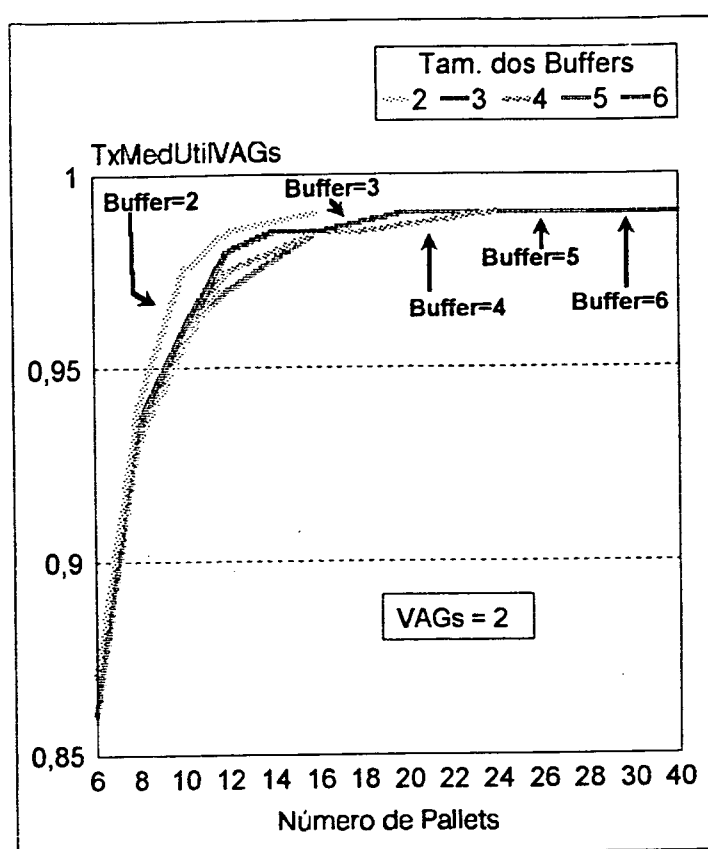


Gráfico 4.15: Resultados para 2 VAGs

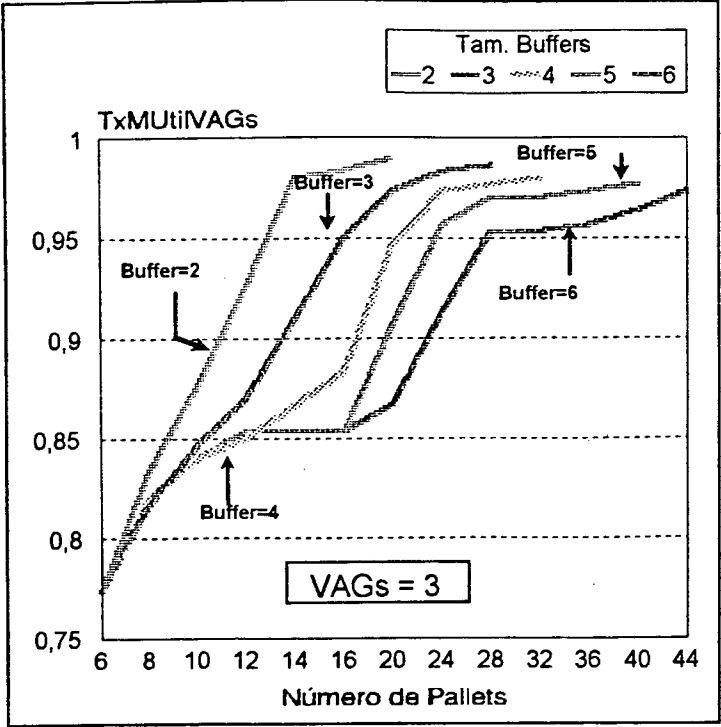


Gráfico 4.16: Resultados para 3 VAGs

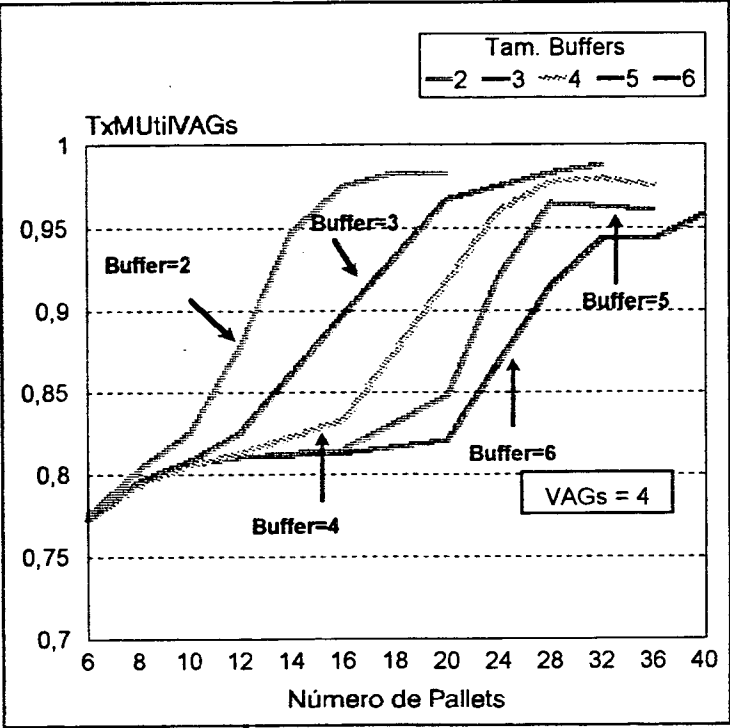


Gráfico 4.17: Resultados para 4 VAGs

4.4 Sumário

Neste capítulo apresentamos e analisamos os resultados de mais de 500 experimentos de simulações e testes, envolvendo o modelo de simulação desenvolvido no capítulo III. O propósito destes testes foi assegurar a validade do modelo, demonstrar sua contribuição para a obtenção de resultados promissores no encaminhamento do problema de planejamento e operação de SFM e, permitir-nos um maior conhecimento dos efeitos causados pelos diversos fatores analisados, sobre o desempenho operacional destes sistemas de manufatura.

Os resultados e conhecimentos obtidos, principalmente sobre as influências que as políticas de gerenciamento operacional (regras de despacho de peças e veículos) exercem sobre o desempenho do sistema, bem como suas combinações com outras variáveis e parâmetros dos SFM (tamanho dos buffers, número de pallets e número de VAGs) nos levam a ampliar o modelo de simulação desenvolvido.

Para tal, propomos a incorporação de um sistema especialista (SE) àquele modelo. Este sistema especialista, utilizando-se dos conhecimentos adquiridos nas simulações realizadas, deverá buscar, de forma interativa, uma combinação entre as diversas variáveis operacionais analisadas e os objetivos de produção desejados, considerando prioritariamente um desempenho aceitável do SFM.

CAPÍTULO V

ESTRUTURA DO PROTÓTIPO DO SISTEMA INTELIGENTE DE SIMULAÇÃO

No capítulo anterior, os vários elementos associados ao planejamento e controle operacional de SFM, e suas combinações, foram estudados e testados. Uma série de conclusões sobre seus efeitos foram delineadas. A partir destes estudos, testes e conclusões, um conjunto organizado de conhecimentos foi adquirido. Tais conhecimentos são parte integrante de um agregado maior de informações, regras e procedimentos que, aliados ao bom senso, são utilizados pelos analistas na busca do melhor desempenho operacional possível em SFM.

Neste capítulo, apresentamos a estrutura do protótipo de um sistema inteligente de simulação que constitui-se da integração entre o modelo de simulação com alocação dinâmica de recursos, desenvolvido no capítulo III, e um sistema especialista.

O sistema especialista procura reunir o conhecimento por nós adquirido quando da aplicação do modelo aos vários problemas apresentados no capítulo IV além, de outras informações, regras e procedimentos gerais, tomados da literatura e do contato com profissionais da área.

5.1 Objetivo do Protótipo

O protótipo desenvolvido, tem por objetivo auxiliar o usuário/decisor a buscar um desempenho operacional satisfatório para o SFM sob análise, considerando os seus inúmeros parâmetros e restrições operacionais.

O uso de um modelo de simulação, ou mesmo de outras ferramentas utilizadas na busca de soluções que satisfaçam os objetivos, as metas e as várias restrições, é uma tarefa com um alto grau de dificuldade imposta a qualquer decisor.

Pelo uso do protótipo, o usuário poderá desenvolver uma série de exercícios interativos/investigativos que busquem alcançar o equilíbrio operacional do SFM sobre o qual ele está trabalhando.

5.2 Estrutura do Protótipo

O protótipo é constituído de uma estrutura modular, visando facilitar seu uso. Os módulos, que possuem diferentes características e funções, são acessados por meio de uma interface gráfica, pela ação direta do usuário. A figura 5.1, apresenta a estrutura geral do protótipo. Nela, observamos seus três principais elementos: o Módulo de Interface Gráfica Usuário/Sistema (GUI), o Módulo de Simulação (MS) e o Módulo de Análise (MA). O primeiro é a interface gráfica que permite ao usuário a interação entre ele e o sistema, com pleno acesso aos dois outros módulos. A ativação, via GUI, do MS, torna possível ao usuário a execução das tarefas associadas a modelagem e simulação de um SFM. A ativação, também via GUI, do MA, permite ao usuário, valendo-se do auxílio do sistema especialista, realizar a análise dos resultados das simulações e a formulação de novas estratégias gerenciais, afim de melhorar o desempenho operacional do SFM.

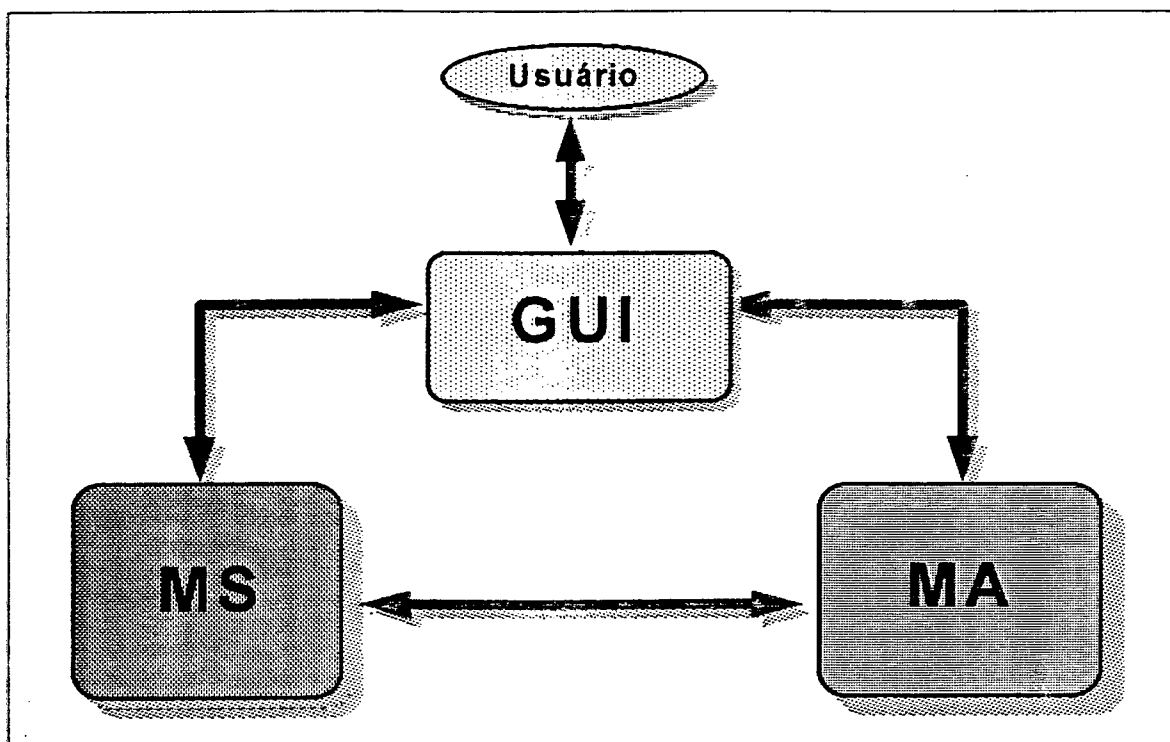


Figura 5.1: Estrutura geral protótipo

5.2.1 Interface Gráfica Usuário-Sistema (GUI)

O usuário tem no módulo de interface gráfica (GUI), o principal meio de acesso aos dois módulos executores de tarefas do protótipo, o MS e o MA. Além desta função básica, é também via GUI, que o usuário executa o processo iterativo na busca de um melhor desempenho operacional do SFM. Neste processo, são executadas trocas de informações entre o MS e o MA. A comunicação entre os dois módulos operativos fica transparente ao usuário que, no entanto, permanece no controle de todo o processo via interface gráfica.

A interface gráfica do usuário (GUI), é constituída de um conjunto de objetos gráficos, tais como telas, janelas, menus, botões, etc. ,através da qual o usuário controla todas as ações executadas pelo protótipo. A tela inicial da interface, apresentada na figura 5.2, permite ao usuário, por meio dos botões, acessar tanto o MS quanto o MA, dando início a trabalhos de simulação, análise ou a ambos de forma interativa.



Figura 5.2: Tela inicial do protótipo com alguns dos elementos da interface gráfica (GUI)

5.2.2 Módulo de Simulação

O modelo de simulação desenvolvido neste trabalho, faz uso do SIMAN [PENG DEN, 1990], uma linguagem de simulação (programação) voltada para este tipo de aplicação.

Linguagens de simulação, assim como algumas linguagens de programação, permitem relativa flexibilidade na abordagem ao problema mas, muitas vezes, são ineficientes na criação de interfaces com o usuário. Esta ineficiência, geralmente não causa grandes transtornos quando usuário e programador são a mesma pessoa. No entanto, quando se trata de diferentes pessoas, a necessidade de interfaces amigáveis é quase que uma obrigação.

A partir desta constatação, desenvolvemos uma interface gráfica voltada ao MS, para que o usuário possa navegar e fazer uso deste com certa facilidade.

O módulo de simulação (MS) deve ser ativado pelo usuário sempre que este desejar executar a simulação de um modelo de SFM ou outras tarefas associadas a uma simulação, como por exemplo, visualizar e/ou trocar parâmetros de todos os elementos constituintes de um SFM, visualizar e/ou trocar parâmetros relativos as regras ou políticas operacionais, visualizar e/ou trocar parâmetros relativos ao cenário sob o qual deseja testar/simular o sistema, salvar os parâmetros em arquivos para utilização futura e, executar a simulação.

Uma vez que o usuário tenha ativado o MS, duas alternativas de ações básicas, como pode ser visto na figura 5.3, podem ser ativadas: a execução da simulação ou a definição de parâmetros, para a posterior execução de uma simulação.

A opção Executar Simulação permite que o usuário execute diretamente do protótipo uma simulação de um modelo de SFM cujos parâmetros tenham sido previamente definidos e salvos em arquivo.



Figura 5.3: Tela inicial do Módulo de Simulação com suas duas opções

Já a opção Definir Parâmetros da Simulação, leva o usuário as alternativas Mostrar/Alterar Parâmetros ou Salvar Parâmetros. A primeira alternativa, permite que sejam verificados ou alterados os parâmetros relativos aos principais elementos de um modelo de SFM tais como:

- As peças a serem produzidas e todas as suas características, isto é, suas seqüências de operações e respectivos tempos;
- As máquinas do SFM (número de máquinas, operações executáveis, índices de eficiência, os tamanhos de buffer, etc.);
- Os equipamentos de transporte, VAGs e Pallets (número disponíveis, velocidade dos VAGs, etc.);
- As políticas operacionais desejadas ao despacho das peças, dos veículos, etc;
- O cenário sob o qual a fabricação será simulada, caracterizado pelos tipos de peças a serem produzidas, o tamanho do lote e seu mix.

Ao terminar a verificação ou alteração dos parâmetros, o usuário poderá passar a execução direta do modelo ou salvar as alterações em arquivos para posterior utilização.

A figura 5.4 apresenta uma visão geral das possibilidades oferecidas ao usuário/decisor quando este ativa o MS.

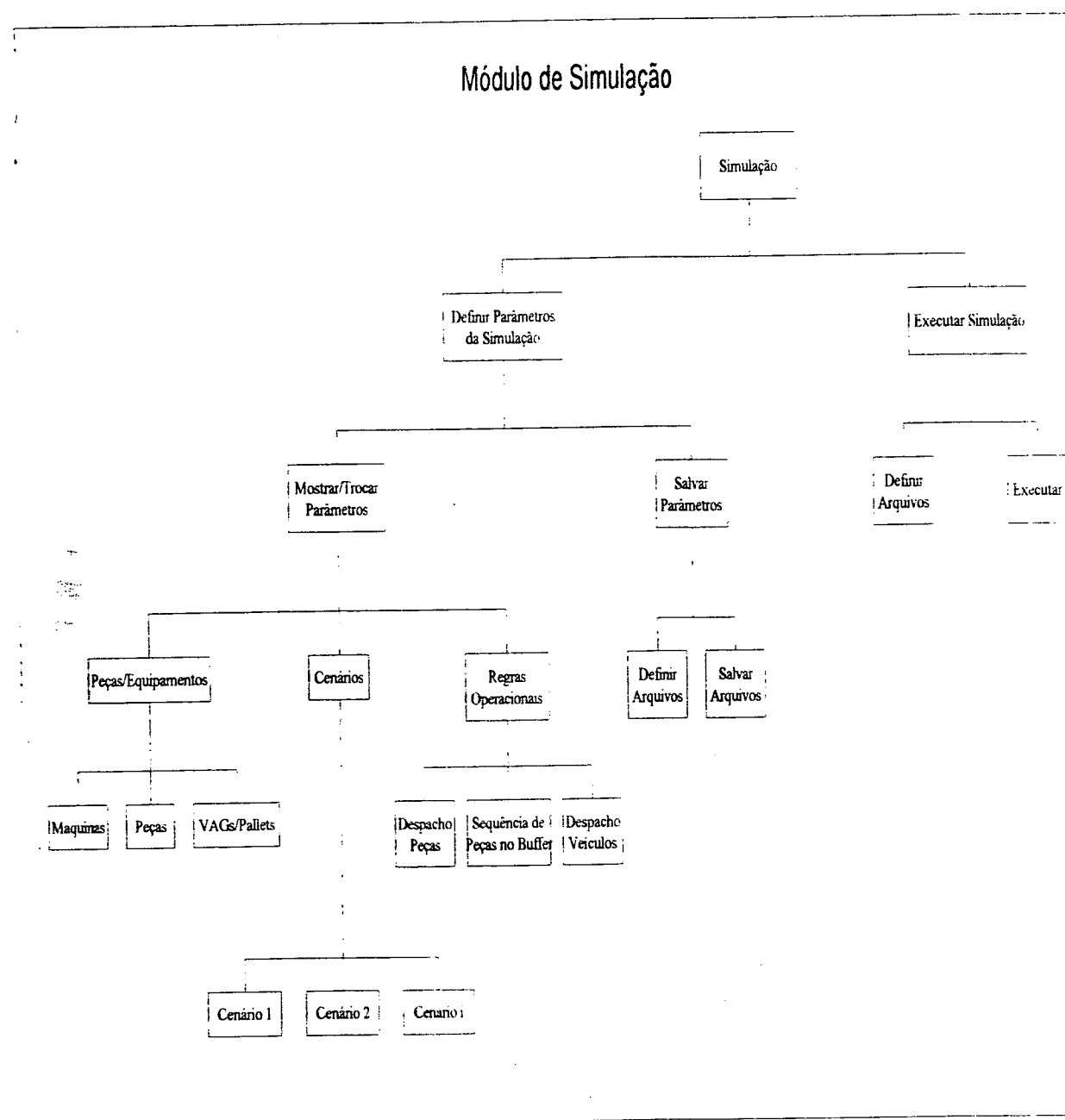


Figura 5.4: Estrutura do Módulo de Simulação

5.2.3 Módulo de Análise

A partir da tela inicial do protótipo (ver figura 5.2), o usuário poderá ativar o Módulo de Análise dos Resultados da Simulação (MA). Este módulo, permite três opções básicas, como se pode observar na figura 5.5 que apresenta a tela inicial do MA.

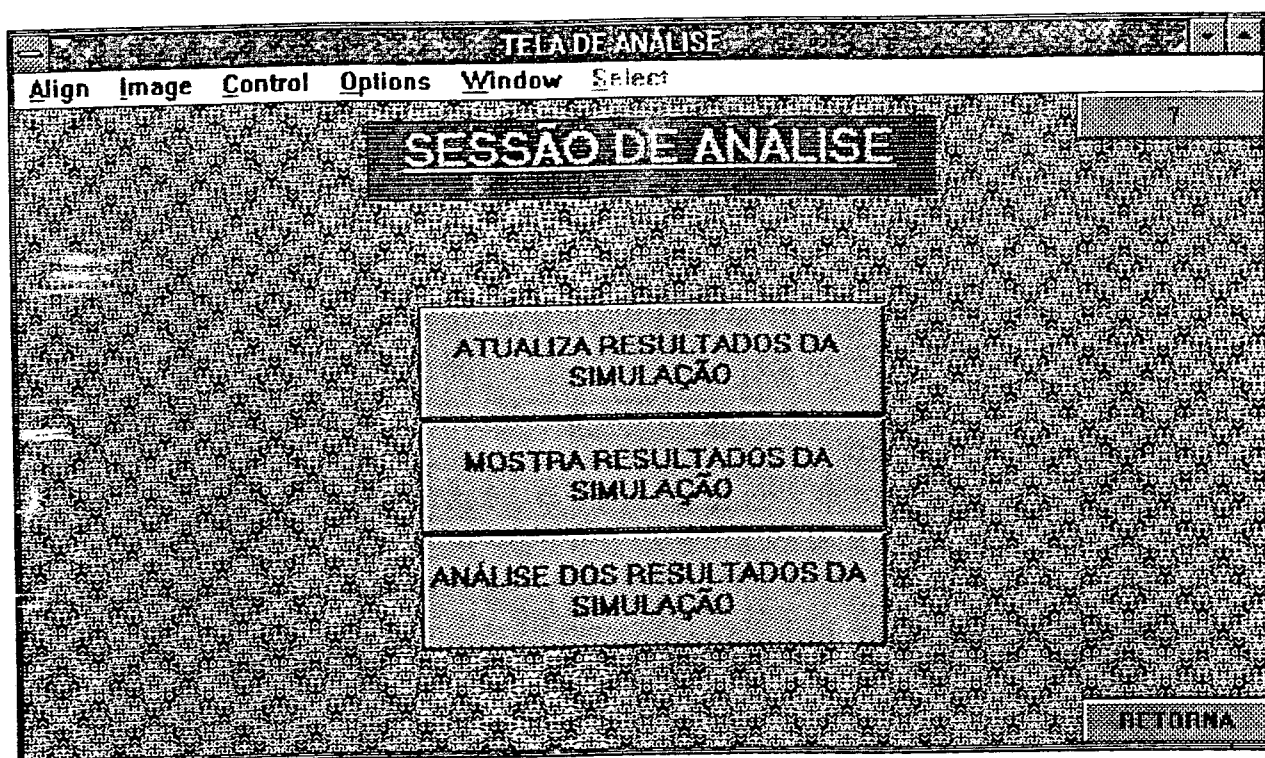


Figura 5.5: Tela inicial do Módulo de Análise

Uma vez nesta janela, a escolha da primeira opção (Atualiza Resultados da Simulação) permite ao usuário atualizar os arquivos com os resultados da última simulação. Desta forma, após uma rodada de simulação, o usuário deve primeiramente atualizar os resultados pois, de outra forma, se optar diretamente pela sua visualização (segunda opção), estará observando os resultados da simulação anterior. Por opção do usuário, esta atualização poderá ser automática após cada simulação.

A escolha da segunda opção (Mostra Resultados da Simulação) permite ao usuário visualizar os resultados de uma simulação. Por exemplo, nos resultados relativos as peças manufaturadas, o analista poderá verificar quantas peças de cada tipo foram produzidas, se as metas de produção foram alcançadas, qual o tempo médio de fluxo destas pelo

sistema ou ainda um diagnóstico deste tempo médio, estabelecido de acordo com parâmetros anteriormente definidos. A figura 5.6 ilustra estes resultados.

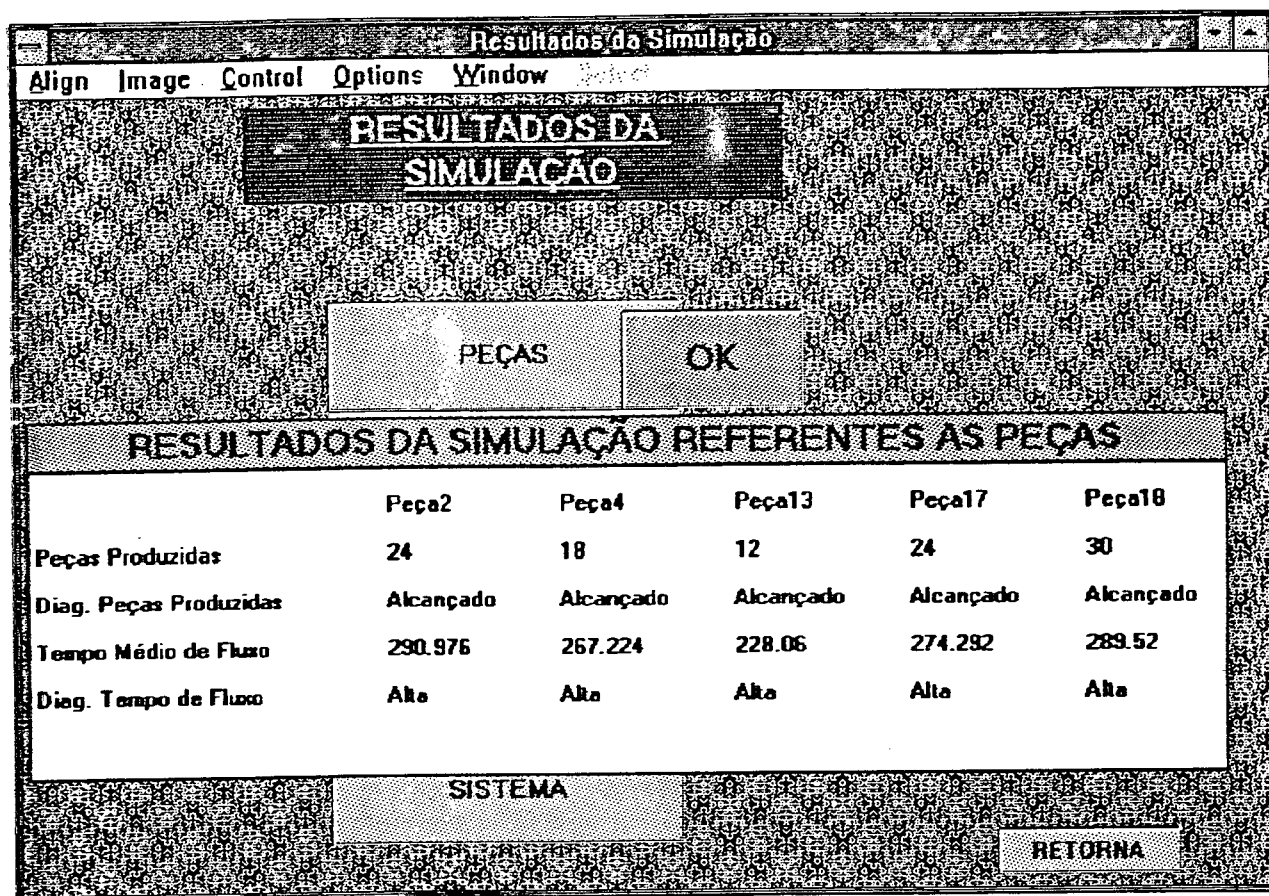


Figura 5.6: Janela com os resultados da simulação referentes as peças produzidas

A terceira opção (Análise dos Resultados da Simulação), conduz o usuário ao principal objetivo do MA, isto é, inicializar o processo de análise dos resultados. Esta opção ativa o SE e toda a estrutura de decisão (ver detalhes no item 5.3), possibilitando ao usuário contar com um auxiliar para a elaboração de análises relativas ao desempenho operacional do SFM.

A figura 5.7 mostra a estrutura geral do módulo de análise e as possibilidades oferecidas ao analista quando este o ativa.

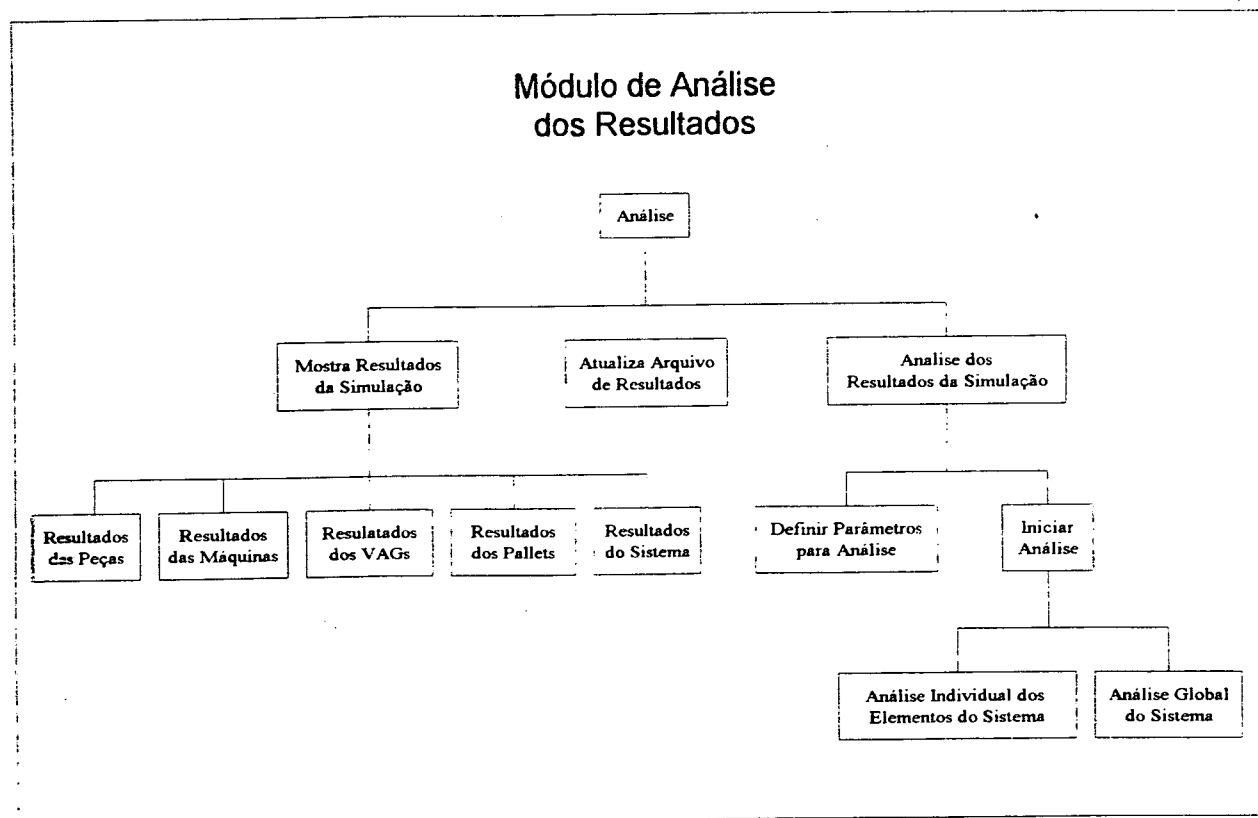


Figura 5.7: Estrutura do Módulo de Análise dos Resultados da Simulação

5.3 O Sistema de Decisão

Nos itens anteriores detalhamos a estrutura do protótipo, descrevendo seus três principais módulos, em especial o ambiente gráfico pelo qual o usuário interage com o sistema.

O objetivo dos tópicos seguintes é descrever os detalhes internos do protótipo, isto é, os processos que se desencadeiam quando o usuário inicia e dá continuidade a todo um procedimento interativo na busca de um desempenho operacional satisfatório para o SFM sob análise.

Iniciamos detalhando o processo de comunicação entre o MS e o MA, parte integrante do procedimento interativo/investigativo, e concluímos com uma descrição da estrutura de decisão, em especial do sistema especialista.

5.3.1 Comunicação entre o Módulo de Simulação e o Módulo de Análise

Como pode ser visto na figura 5.8, os dois principais módulos do protótipo, o MS e o MA, comunicam-se entre si, tornando possível ao analista a execução automática das simulações e análises, sem precisar estar saindo ou entrando em diferentes programas computacionais.

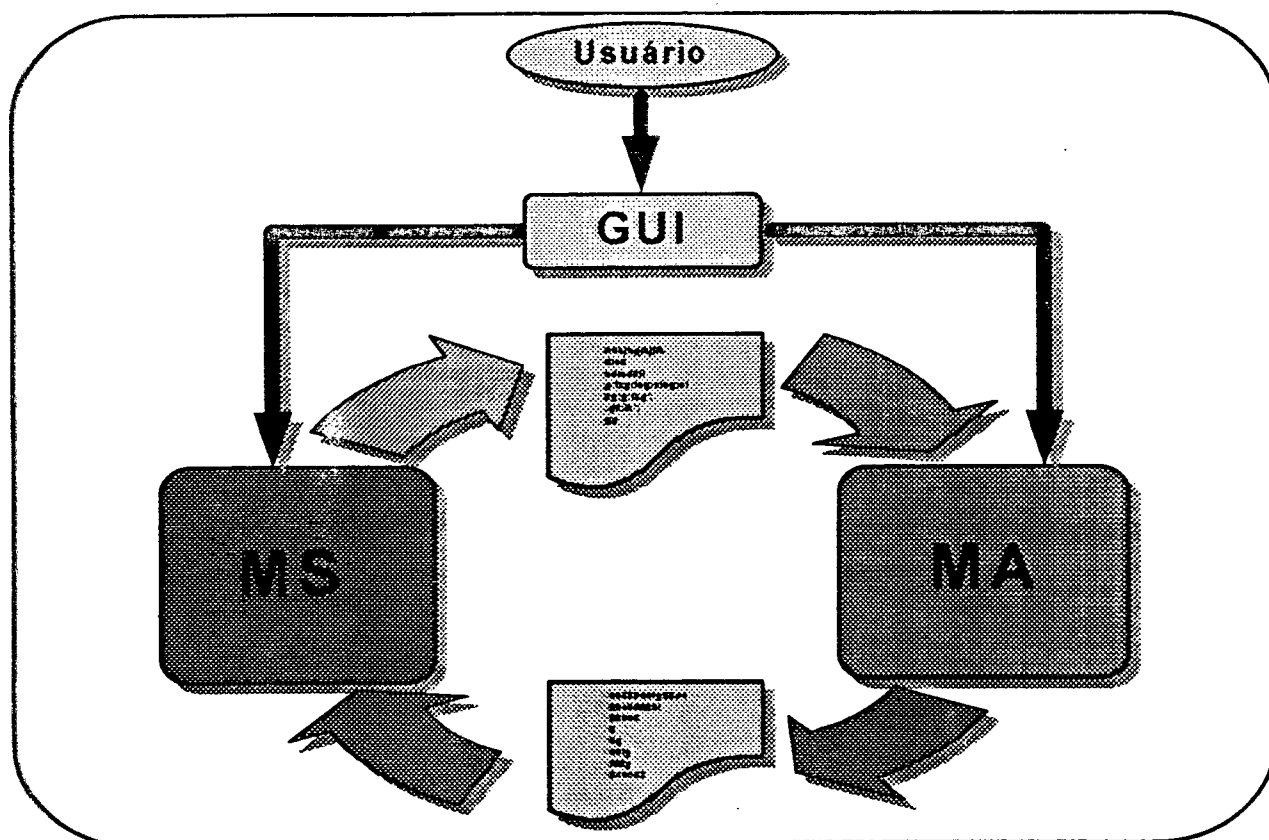


Figura 5.8: Processo de comunicação entre os Módulos de Simulação e Análise por meio de arquivos de dados

Toda a comunicação entre os dois módulos é feita por meio de funções que criam, exportam e importam arquivos, coordenando a troca de dados entre os módulos. Para poder executar uma simulação, o MS deve ser alimentado com todos os parâmetros operacionais necessários.

Uma vez que o modelo de simulação está totalmente parametrizado, ao dar início a uma simulação, um programa conectado ao modelo busca um arquivo contendo todos os

parâmetros. Este arquivo foi previamente construído pelo analista, fazendo uso da interface gráfica do usuário (GUI), e exportado ao MS.

Após o término da simulação, o MS inicia um processo no sentido inverso, isto é, cria outro arquivo de dados, contendo todos os elementos necessários à análise dos resultados, e o envia de volta ao MA. O analista pode então dar início aos processos de análise.

Na figura 5.9, observa-se com mais detalhes, os elementos das duas estruturas.

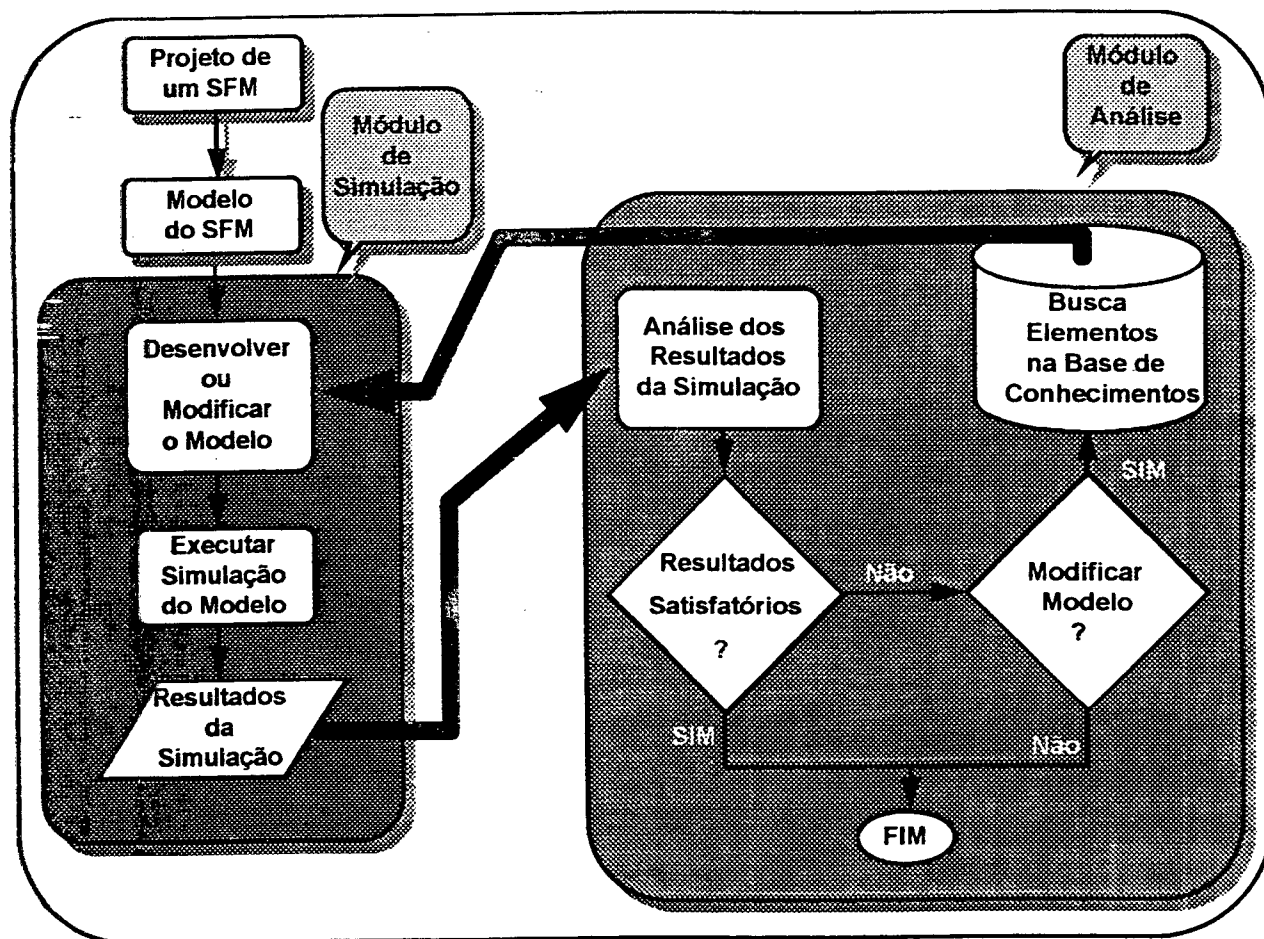


Figura 5.9: Elementos básicos do MS e do MA

O desenvolvimento deste tipo protótipo, em especial a construção de interfaces entre seus vários elementos é, geralmente, facilitado pelo uso de técnicas de Programação Orientada a Objeto (POO).

Com este propósito, fizemos uso de uma ferramenta especialmente voltada ao desenvolvimento de sistemas inteligentes com orientação a objeto, o KAPPA PC Development Applications System [INTELLICORP, 1992], para a construção do protótipo.

O KAPPA é uma ferramenta híbrida que permite a representação do conhecimento sob diferentes formas e, vários mecanismos de controle na máquina de inferência. Além disso, por rodar sob o ambiente operacional Windows 3.1, oferece as mesmas possibilidades de criação de interfaces gráficas com o usuário daquele ambiente. Suas características são apresentadas a seguir, em conjunto com os principais elementos constituintes do sistema de decisão, com suas regras de produção, frames, métodos e funções

5.3.2 A Estrutura de Decisão

A estrutura de decisão do protótipo foi montada de acordo com os elementos, próprios para o desenvolvimento de aplicações, disponíveis no KAPPA. Fizemos intenso uso de estruturas voltadas ao desenvolvimento de sistemas especialistas, conceituadas no capítulo 2, tais como classes, objetos, frames, regras de produção, etc. (ver item 2.6.3. daquele capítulo), as quais são agora revistas, mostrando-se seu emprego.

5.3.2.1 Classes, Objetos, Frames, Slots e Métodos

No KAPPA, à semelhança de outras linguagens de POO (como C++, SmallTalk, ADA, etc.), o sistema é estruturado por meio de classes e objetos, o que permite o uso do conceito de hierarquia, isto é, os atributos pertencentes a um objeto, são herdados pelos seus descendentes (subclasses, instâncias, etc.).

No presente estudo, objetos como máquinas, transportadores, peças, operações, etc., são representados em classes. Cada uma possui um conjunto de atributos para descrevê-los e qualquer dos seus herdeiros (subclasses ou instâncias), tais como Máq1, Máq2, VAGs, Peça1, etc, também os possuirão. Além disso, alguns destes herdeiros possuem características próprias, que os diferenciam uns dos outros.

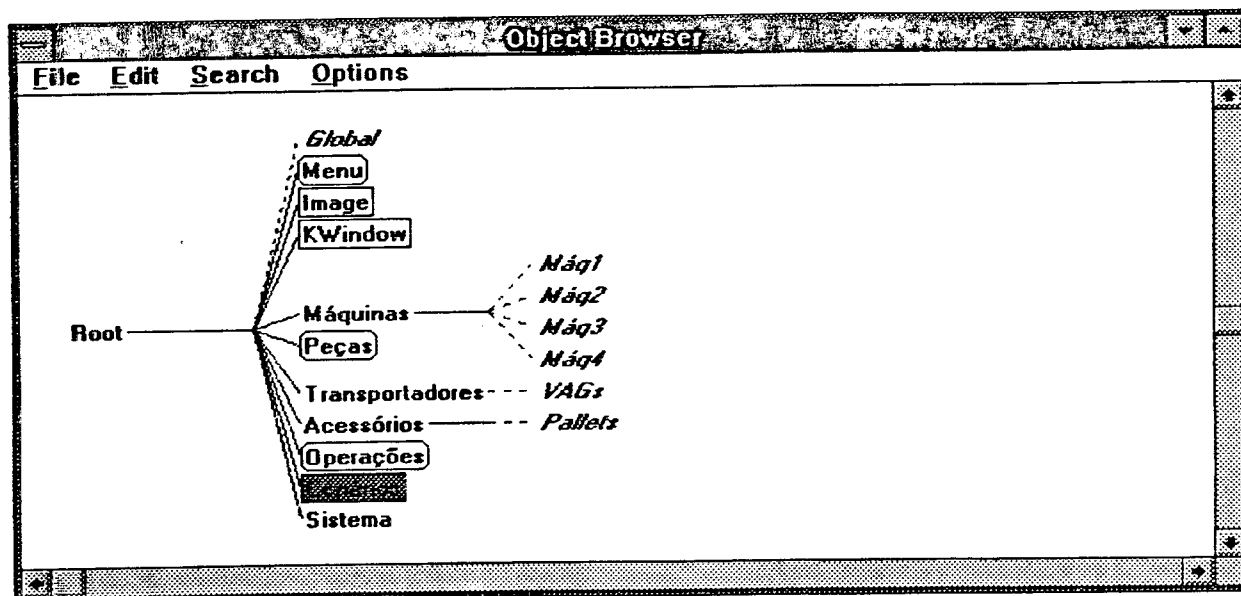


Figura 5.10: Algumas classes e instâncias do protótipo

A figura 5.10 ilustra parte da estrutura do protótipo, apresentando alguns dos seus objetos na forma de classes e instâncias. Podemos observar, por exemplo, a classe Máquinas e seus descendentes Máq1, Máq2, Máq3 e Máq4. Os últimos são chamados de instâncias, por se encontrarem no último nível desta classe.

Aliado ao conceito de classes, está o conceito de *frames* (ver item 2.6.3.1, do capítulo 2), elemento típico de sistemas inteligentes orientados a objeto. Os frames são as estruturas nas quais estão definidos os objetos do sistema, bem como a forma e os meios pelos quais eles interagem entre si e com o mundo exterior, por meio do envio de mensagens.

Um frame constitui-se de dois tipos básicos de informações: um conjunto de atributos, também conhecidos como *slots*, e um conjunto de métodos. Portanto, cada objeto definido no sistema (classes, subclasses e instâncias), possui associado si, dentro de um frame, um conjunto de atributos e um conjunto de métodos associados a estes atributos.

Os atributos são tipicamente um conjunto de características, ou qualidades de cada objeto, através dos quais podemos distingui-los uns dos outros. Já os métodos, são semelhantes a funções ou rotinas, designados a executar ou desencadear uma ação. Estão associados a um ou mais atributos, sendo automaticamente ativados quando ocorre a alteração dos

valores destes atributos. Estes dois conjuntos de informações contidas nos frames, caracterizam, de maneira completa, cada objeto do sistema.

Na figura 5.11 podemos observar a classe Máquinas com a lista parcial de seus atributos (slots) e dois métodos pertencentes a esta classe e a seus descendentes.

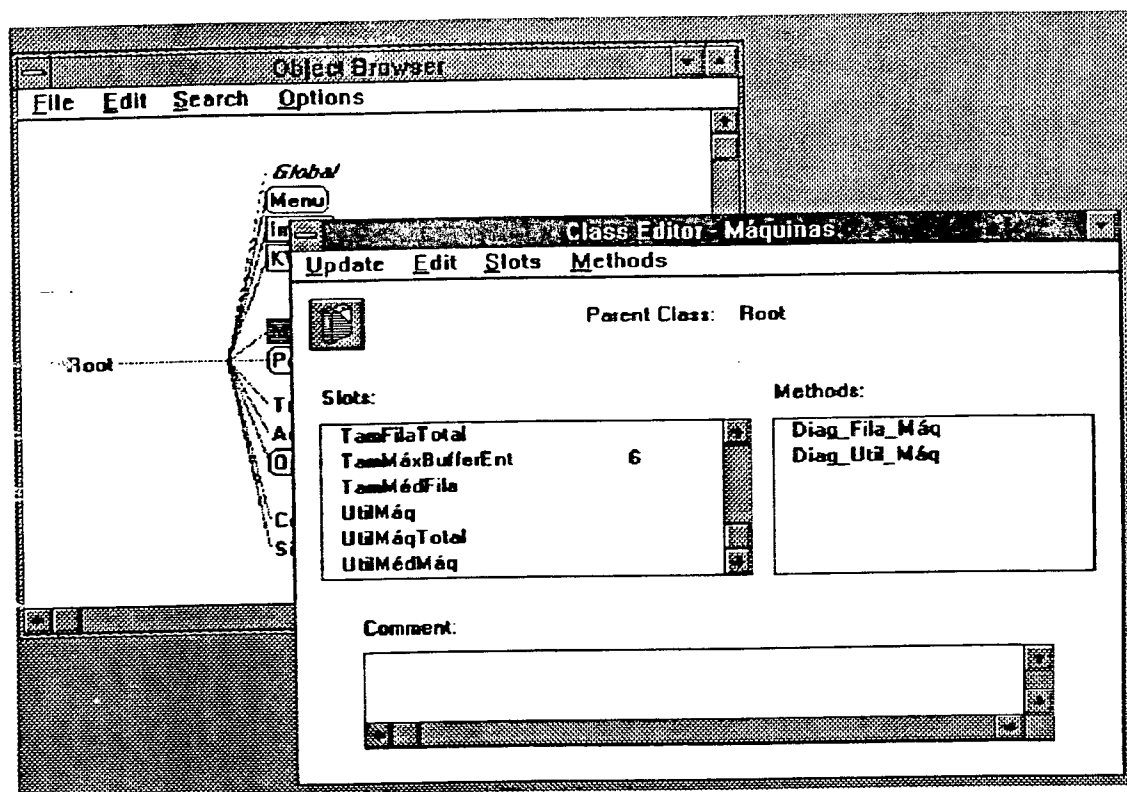


Figura 5.11: A classe de Máquinas exibindo a lista parcial de seus atributos e a lista de métodos associados a alguns deles

Nos sistemas desenvolvidos com o KAPPA, os métodos podem ser ativados de três diferentes maneiras:

- por meio de mensagens enviadas por outros método, pertencentes a outros objetos;
- através da alteração do valor de um slot ou atributo ao qual o método esteja atrelado;
- via funções, que também podem enviar mensagens para ativar os métodos.

Uma vez que a metodologia proposta pressupõe um processo interativo, com base na comunicação entre o módulo de simulação e o módulo de análise, a possibilidade do uso de métodos, associados aos atributos dos objetos do sistema, permite automatizar, sobremaneira, este processo de comunicação.

A automatização deste processo, traz consigo uma série de facilidades para o usuário uma vez que inúmeros procedimentos tais como: criação, leitura e atualização de arquivos, análise de resultados, atualização de parâmetros para novas rodadas de simulação, etc, são repetitivos e constantemente realizados.

Um exemplo típico da aplicação de métodos para a automatização de procedimentos, ocorre ao término das rodadas de simulação, durante as interações. Naquele momento, inúmeras variáveis devem ter seus resultados atualizados e avaliados. Dentre estas podemos citar, por exemplo, a taxa média de utilização das máquinas.

O valor final destas taxas é calculado no programa de simulação, atribuído as variáveis correspondentes a cada máquina e transferido ao arquivo de resultados, que será lido pelo módulo de análise. No módulo de análise, a leitura do arquivo de resultados da simulação, atualiza os valores de vários atributos de objetos do sistema.

Por exemplo, a classe Máquinas, que é constituída de quatro instâncias, Maq1, Maq2, Maq3 e Maq4 (ver figura 5.10), possui um atributo denominado UtilMedMaq. Por força do conceito de hereditariedade, suas quatro instâncias herdam, automaticamente, este atributo. Desta forma, quando ocorre a leitura do arquivo de resultados, os valores do atributo UtilMedMaq, nas quatro instâncias da classe Máquinas, são atualizados, recebendo os valores da taxa média de utilização de cada uma das máquinas do SFM simulado.

Conforme citamos, a alteração ou atualização do valor de um slot ou atributo, pode ativar um ou mais métodos que estejam a ele associados. Na classe das Máquinas e, conseqüentemente, nos seus decendentes (instâncias), o atributo UtilMedMaq está associado ao método Diag_Util_Máq. Desta forma, sempre que o Módulo de Análise executa a leitura do arquivo de resultados, com a imediata atualização do atributo UtilMedMaq, uma mensagem é, internamente, disparada, ativando o método Diag_Util_Máq.

Na figura 5.12 observamos o método Diag_Util_Máq que é ativado pela atualização dos valores do atributo UtilMedMáq. Na janela intermediária, é possível ver o atributo Diag_Util_Máq que recebe o resultado da execução do método.

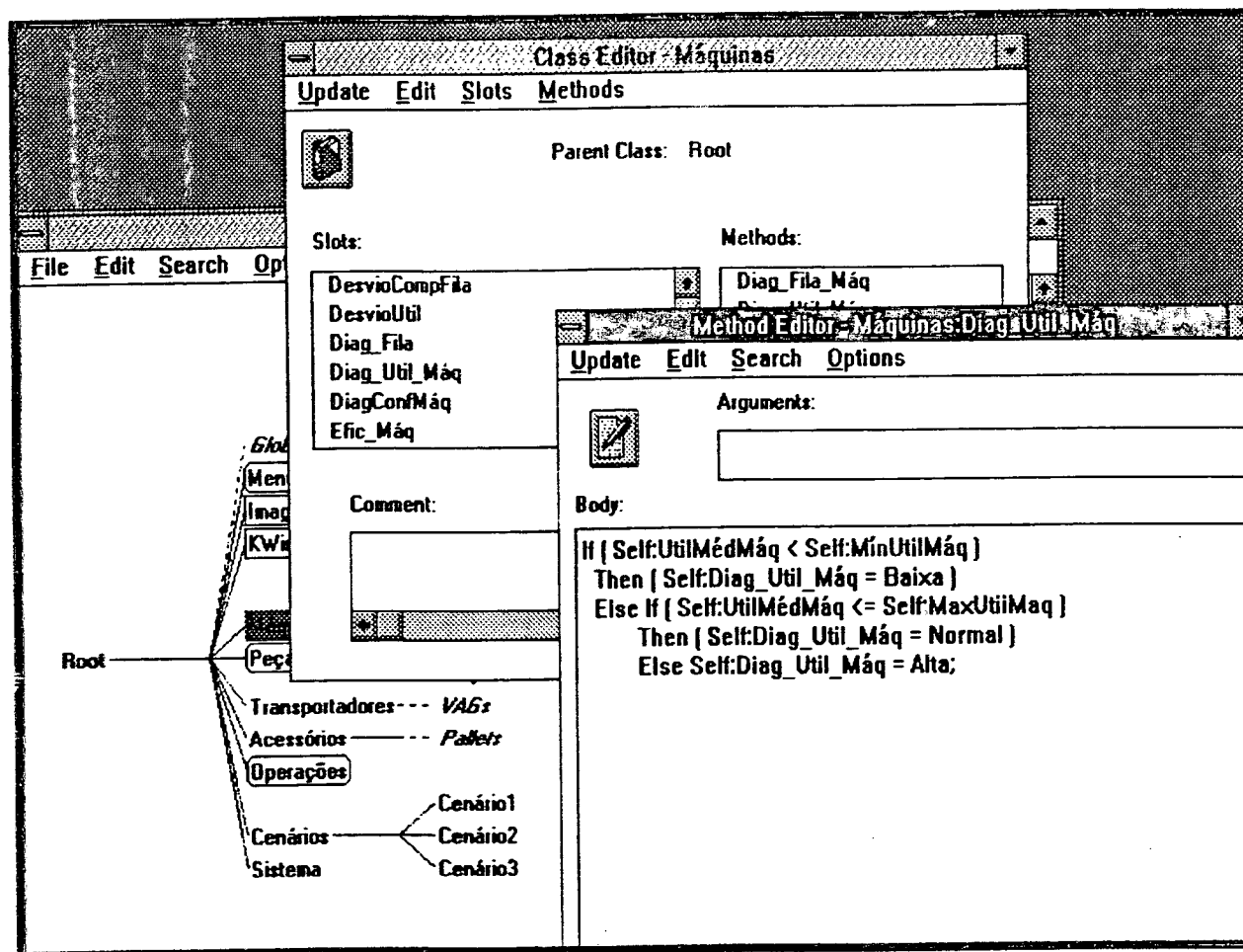


Figura 5.12: O Método Diag_Util_Maq, associado ao slot UtilMedMáq

O objetivo deste método, em particular, é diagnosticar a forma de utilização das máquinas, isto é, determinar, segundo critérios pré-estabelecidos associados aos atributos MínUtilMáq e MaxUtilMáq, se as máquinas estão sendo sub, sobre ou normalmente utilizadas. Os três possíveis resultados para o atributo Diag_Util_Maq podem ser Baixa, Alta e Normal, respectivamente.

Dependendo do resultado deste diagnóstico, outras mensagens, dirigidas a outros métodos, podem ser enviadas, dando início a um processo em cadeia, o qual é automático

e transparente ao usuário, mas que deixa o sistema sempre atualizado ao fim de cada interação.

A estrutura completa das classes, com seus slots e métodos desenvolvidos no protótipo pode ser vista no Anexo 2, Objetos do Sistema.

5.3.2.2 Regras de Produção

O segundo conjunto de elementos utilizados no desenvolvimento do módulo de análise, é constituído por regras de produção. No KAPPA, existe um ambiente com ferramental completo para o desenvolvimento e emprego de regras de produção. No protótipo utilizamos esta estrutura para desenvolver dois diferentes grupos de regras: o GRDLME (Grupo de Regras para Diagnóstico Local de Máquinas e Equipamentos) e o GRDGME (Grupo de Regras para Diagnóstico Geral de Máquinas e Equipamentos).

No primeiro grupo de regras, incluímos aquelas cujo objetivo principal é o diagnóstico local dos vários grupos de elementos do sistema, tais como, as máquinas, os transportadores, etc.. Assim, se for seu desejo, o analista poderá valer-se destas regras que, em conjunto com alguns métodos, permitem, rapidamente, o diagnóstico do comportamento destes grupos de elementos.

Por exemplo, após cada rodada de simulação, os resultados são automaticamente transferidos ao MA, alimentando e atualizando, os vários pares de objetos: atributos do sistema. Como vimos anteriormente, tais atualizações geram um conjunto de mensagens endereçadas a vários métodos. Estes, uma vez ativados, realizam alguns cálculos e pequenos diagnósticos individuais, isto é, ao nível de cada equipamento, máquina, transportador, tipo de peça fabricada, etc.

Este conjunto de novos fatos, decorrentes de cada nova rodada de simulação, pode então ser confrontado com a base de conhecimentos do sistema especialista, mais precisamente com suas regras de produção e, por meio da máquina de inferência do sistema, gerar diagnósticos de maior amplitude. Tais diagnósticos podem indicar, por exemplo, gargalos no sistema produtivo e, partir daí, sugerir ações para evita-los.

No segundo grupo de regras, incluem-se aquelas que tratam da análise global do desempenho do SFM. Esta análise, objetiva o seu balanceamento operacional.

Após definir claramente para o sistema suas preferências em termos dos parâmetros e regras operacionais, o analista pode dar início ao processo interativo de simulação e análise. A cada simulação executada, o sistema realiza automaticamente os diagnósticos individuais, utilizando os métodos e algumas das regras do primeiro grupo. Nesta primeira fase não há interferência do usuário. A partir deste ponto porém, por meio deste segundo grupo de regras, o usuário, se assim o desejar, poderá ser auxiliado na análise do comportamento operacional do sistema. O resultado de cada análise, é uma indicação por parte do MA, do tipo de ação que deve ser efetuada para uma melhoria contínua do desempenho do SFM. O SE contido no MA procura levar a convergência do processo.

O encerramento das análises ocorre quando o MA considera que, dentro dos limites e restrições impostas pelas preferências do usuário, não haverá mais possibilidades de melhorias no desempenho do SFM. No capítulo VI, faz-se uma aplicação do protótipo a um dos problemas vistos no capítulo IV, demonstrando sua maneira de operar, utilizando os dois grupos de regras.

5.3.3 O processo de Análise Executado pelo MA

Internamente ao MA, o processo de análise se inicia com uma varredura sobre os resultados da simulação, gerando uma série de diagnósticos que sintetizam o desempenho dos principais elementos do sistema. Estes diagnósticos são realizados automaticamente pelos diversos métodos. Dentre os vários diagnósticos preliminarmente realizados, os principais são:

- Diagnóstico de Utilização das Máquinas;
- Diagnóstico de Utilização dos VAGs;
- Diagnóstico do Tempo Médio do Ciclo de Produção;
- Diagnóstico dos Tempos Médios de Fluxo das Peças;
- Diagnóstico das Filas das Máquinas;
- Diagnóstico de Confiabilidade de Máquinas e Transportadores;
- Diagnóstico de Utilização Relativa das Máquinas, etc..

Os resultados de todos estes diagnósticos, a nível local, individuais para cada uma das máquinas e equipamentos, são então examinados, em conjunto, pelos dois grupos de regras anteriormente citados.

Desta forma, para apontar, por exemplo o Diagnóstico Operacional Global das Máquinas, o MA aciona o GRDLME, combinando os resultados do Diagnóstico de Utilização das máquinas com o Diagnóstico de Filas, gerando o Diagnóstico Operacional Global das máquinas.

Num segundo passo, combina, usando as regras do GRDGME, este Diagnóstico Operacional, com os de Confiabilidade, Utilização Relativa das Máquinas e o de Tempo de Carga e Descarga, gerando o Diagnóstico Operacional Global das Máquinas. A figura 5.13 sintetiza o caminho percorrido pelo sistema para a obtenção deste diagnóstico global

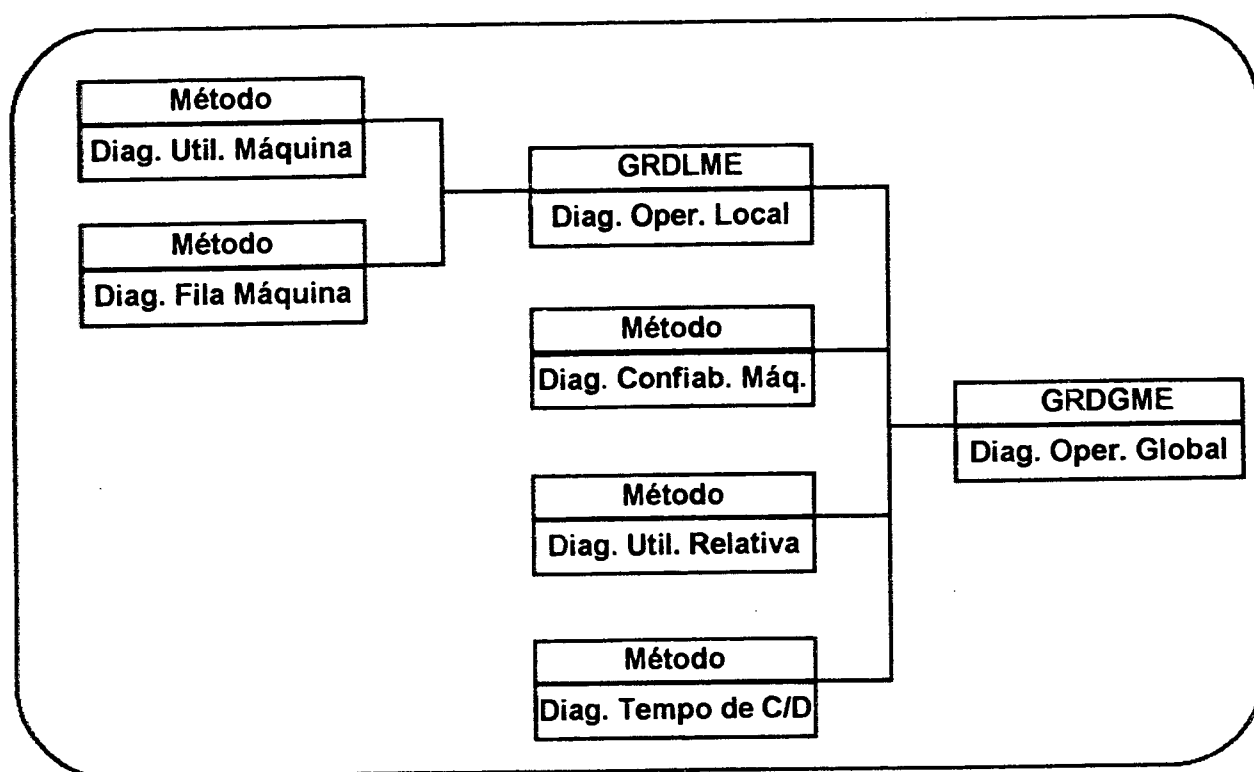


Figura 5.13: Inter-relações entre alguns métodos e conjuntos de regras para a realização do Diagnóstico Operacional Global das Máquinas

De maneira semelhante, outros métodos e os conjuntos de regras concluem diagnósticos para os VAGs, Pallets, Peças, etc.. Conforme colocamos anteriormente, muitos destes métodos e regras, são ativados e atuam de forma transparente para o analista, de tal maneira que, no momento que este decidir buscar auxílio do MA para a tarefa final de análise de desempenho do sistema, estes diagnósticos preliminares já foram determinados, estando o sistema pronto para ativar outros conjuntos de regras e métodos.

O principal conjunto de regras associado a esta fase final das análises, é denominado de ROTMS (Regras de Otimização do Sistema). São elas que, baseando-se em todos os diagnósticos preliminares e, nas rodadas de simulação já realizadas, determinam se o sistema, operacionalmente, está melhorando ou não, sugerindo modificações após cada análise, na busca da obtenção de melhores resultados numa próxima rodada de simulação. A figura 5.14 apresenta um exemplo deste tipo de regra.

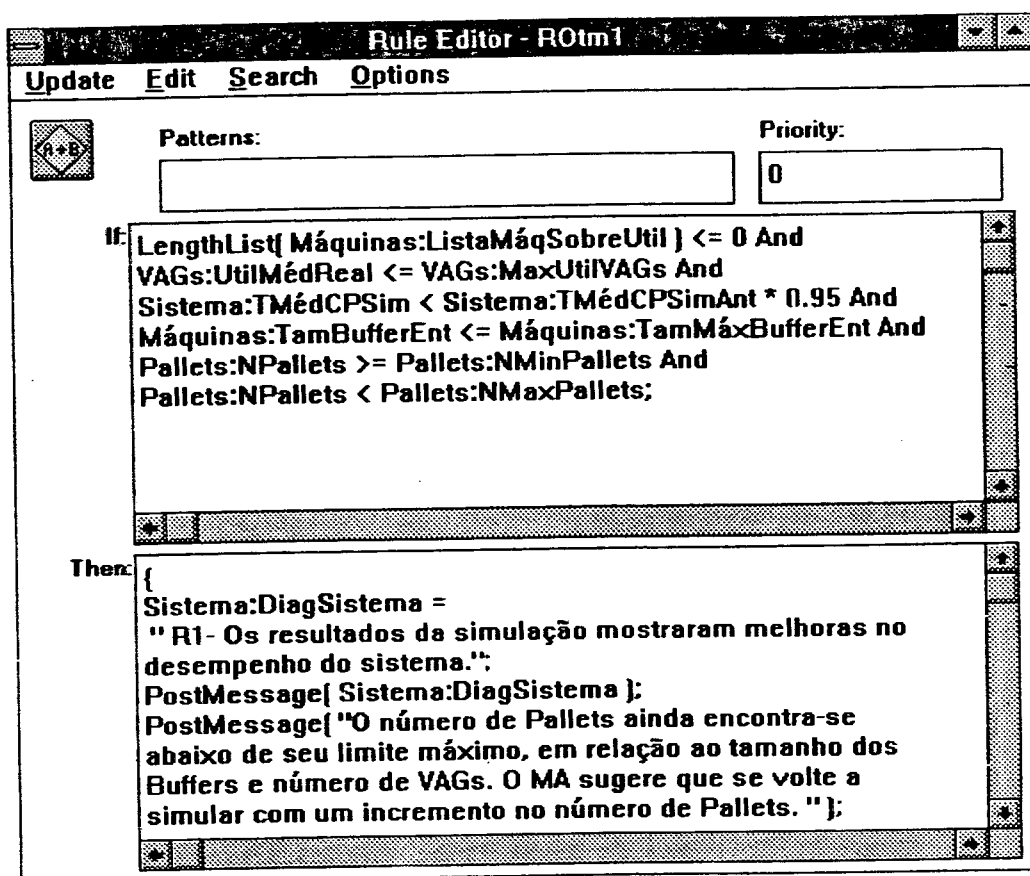


Figura 5.14: Exemplo de uma regra de otimização do sistema

Esta regra, por exemplo, verifica se as seguintes afirmações são verdadeiras: não existe nenhuma máquina sobre utilizada; a utilização média dos VAGs não excedeu o limite máximo estabelecido; houve melhorias sobre o tempo do ciclo de produção anteriormente obtido; o tamanho dos buffers e o número de pallets utilizados, relacionados com o número de VAGs não ultrapassou a relação ótima calculada.

Uma vez que sejam verdades todas as afirmações, a regra diagnostica o sistema e sugere ao usuário uma ação. No caso específico, o Diagnóstico/Ação é :

“Os resultados da simulação mostraram melhoras no desempenho do sistema.

O número de Pallets ainda se encontra abaixo de seu limite máximo em relação ao tamanho dos Buffers e VAGs.

O MA sugere que se volte a simular com um incremento no número de Pallets”

Com base nestes resultados, o usuário pode aceitar a sugestão do sistema e voltar a simular com um incremento do número de pallets ou, realizar outro tipo de modificação de acordo com seu próprio sentimento em relação ao comportamento operacional do sistema.

○ conjunto completo das regras utilizadas no protótipo pode ser encontrado no Anexo 3.

5.4 Sumário

Neste capítulo apresentamos a estrutura de um protótipo de sistema computacional voltado para a simulação e análise de SFM. O objetivo deste protótipo é a integração do modelo de simulação desenvolvido no capítulo III e de um módulo inteligente de análise, composto de uma estrutura híbrida, formada por um sistema especialista, com suas regras de produção e por frames, com suas classes, objetos, instâncias e métodos.

Apresentamos, também, a interface gráfica com o usuário (GUI). Esta interface é o meio que permite, tanto a integração de todos os módulos do protótipo, como a facilidade de seu uso pelo usuário.

No capítulo que se segue, apresentaremos um exemplo de utilização prática do protótipo, demonstrando sua utilidade, flexibilidade e facilidade de uso.

CAPÍTULO VI

APLICAÇÃO E DEMONSTRAÇÃO DE USO DO PROTÓTIPO

O objetivo deste capítulo é demonstrar a utilidade do protótipo desenvolvido e apresentado no capítulo V.

A solução de um problema de planejamento e controle operacional de SFM decorre, usualmente, de um processo interativo, no qual o decisor procura, por meio de experimentos, chegar a um resultado o qual satisfaça suas expectativas, tanto ao nível produtivo, quanto ao nível operacional. No primeiro, o objetivo é o cumprimento das metas de produção, isto é, a manufatura dos lotes de peças dentro de um tempo esperado. No segundo nível, o objetivo é que o alcance das metas de produção seja obtido pôr um sistema operacionalmente estável e balanceado.

A demonstração do uso do protótipo para os fins acima apresentados, será feita descrevendo-se o processo interativo, conduzido pelo decisor, realizado entre o Módulo de Simulação, no qual os experimentos são executados e o Módulo de Análise, onde os resultados são avaliados.

Para fins de uma melhor compreensão, todo o processo será conduzido sobre um problema exemplo já apresentado no capítulo IV. Durante o decorrer da aplicação, faz-se menção aos fatos e regras utilizadas pelo segmento inteligente do protótipo nas análises envolvidas no processo.

O capítulo está subdividido em três seções. Na primeira aborda-se as decisões associadas ao procedimento interativo entre o módulo de simulação e o módulo de análise, conduzido pelo usuário. Na segunda seção, apresenta-se a utilização do protótipo sobre um problema exemplo, mostrando-se de que maneira usuário e sistema interagem e a forma deste último analisar os resultados via sistema especialista. Na terceira e última seção comenta-se o desempenho do protótipo.

6.1 O Procedimento Interativo

O procedimento interativo implica na tomada de decisões em três diferentes estágios como mostra a figura 6.1.

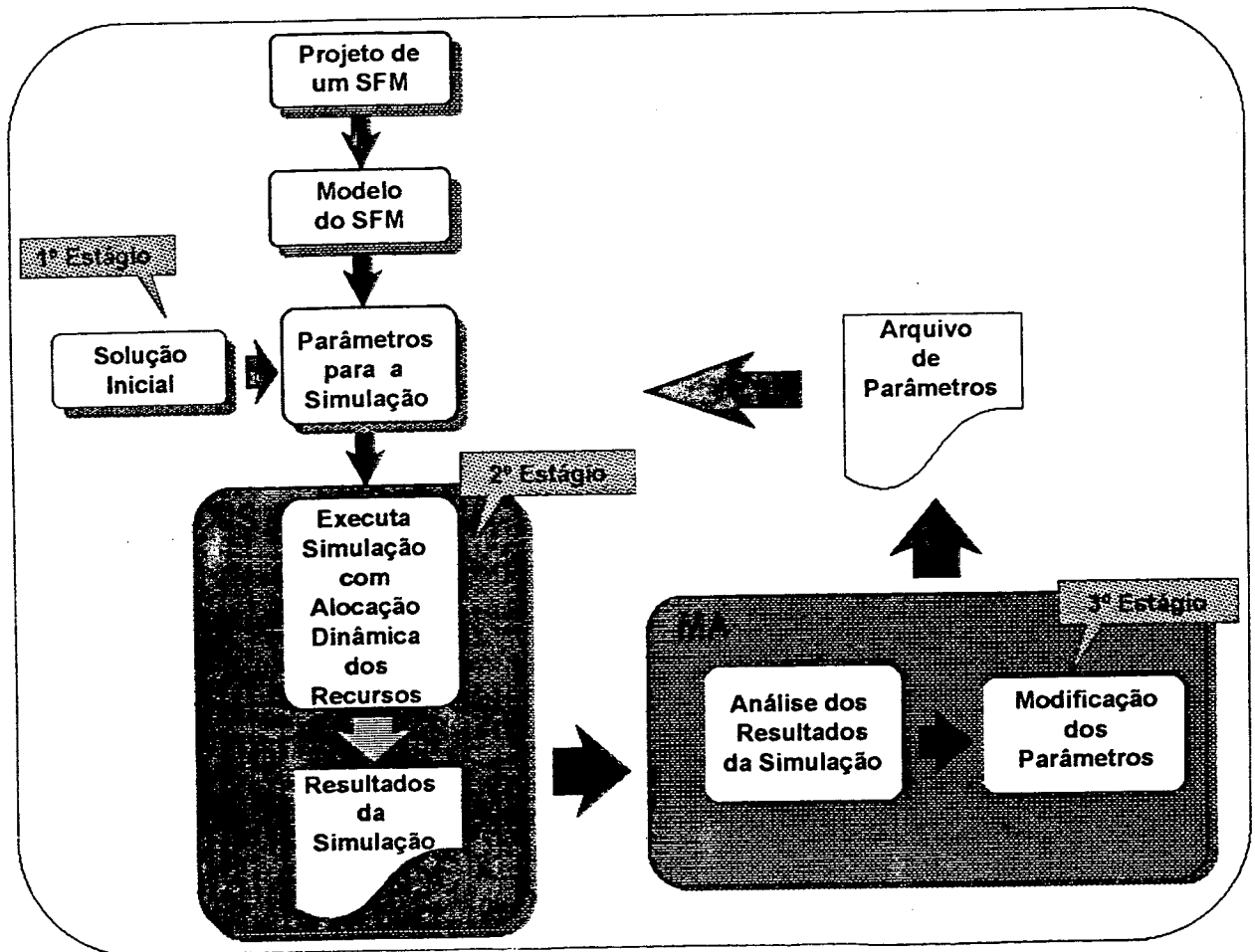


Figura 6.1: Estágios de tomada de decisão no procedimento interativo

O primeiro estágio, trata da adoção de uma solução que dá início ao procedimento interativo. Esta solução, levada a efeito pelo decisor, estabelece os valores mínimos ou de partida, para os diversos parâmetros e políticas operacionais.

A segunda etapa de decisão ocorre durante a alocação dos recursos do sistema. Com a utilização do modelo desenvolvido, esta decisão é tomada durante a simulação da operação do sistema. Vimos, nos capítulos anteriores, que o próprio modelo de simulação

encarrega-se da execução da principal tarefa atribuída ao planejamento da produção, isto é, a alocação dos recursos (máquinas) do sistema, às peças a serem produzidas.

O terceiro estágio de decisão, corresponde ao controle operacional e as modificações paramétricas que devem ser executadas durante o processo de simulação/análise/simulação.

As análises dos resultados das operações realizadas pelo sistema, em cada ciclo simulado, são determinantes sobre as modificações que devem ser realizadas sobre os parâmetros operacionais do modelo/sistema.

Como vimos no capítulo IV, potencialmente, existe um grande número de alternativas para a revisão dos parâmetros quando combinamos o número de veículos transportadores, o número de pallets e o tamanho dos buffers ou seus fatores de utilização. A magnitude das mudanças, bem como a direção destas, requer uma certa dose de julgamento por parte do decisor, ou deste em conjunto com o sistema especialista embutido no MA.

Após a revisão/modificação dos parâmetros em cada interação, o processo de simulação é repetido. Estes procedimentos de simulação/análise/modificação/simulação, repetem-se até que se alcance os resultados de produção esperados, com um razoável balanceamento do sistema produtivo. A estrutura do protótipo, com os vários procedimentos do processo interativo esta ilustrada na fig. 5.9 do capítulo V.

6.2 Aplicação do Protótipo e seus Resultados

No procedimento interativo que agora iniciamos utilizaremos o mesmo modelo de SFM e o mesmo problema exemplo visto no item 4.3.2.2 (Problema Exemplo Utilizado na Comparação de Resultados) do capítulo IV.

6.2.1 Solução Inicial

No item 4.3.4 do mesmo capítulo IV (Busca das Melhores Combinações entre os Elementos Operacionais), realizamos uma série de experimentos, fixando a MTP (Mínima Taxa de Produção) como política para o despacho de peças e FIFO para a política de despacho de veículos. Os demais parâmetros, tamanho dos buffers, número de pallets e

número de veículos foram então combinados gerando uma série de experimentos que foram então analisados sob o ponto de vista de diferentes medidas de desempenho, tais como o Tempo do Ciclo de Produção, Tempo Médio de Fluxo ou de Passagem, Taxa de Utilização de Máquinas, Veículos, etc..

Assim como no capítulo IV, a solução inicial adota MTP e FIFO como políticas operacionais de despacho de peças e veículos, respectivamente. Quanto ao número de veículos, VAGs, embora, nos experimentos realizados, só tenhamos apresentado resultados com 2, 3 e 4 VAGs, a título de ilustração do comportamento do módulo de análise do protótipo, vamos adotar como solução inicial 1 VAG. O tamanho inicial dos buffers será de 2, um para saída e outro para entrada de peças. A definição do número inicial de pallets no sistema é baseada no número de máquinas. Como temos quatro máquinas com buffers de tamanho 2, adotaremos 8 para solução inicial. Desta forma, a solução inicial adotada fica com a seguinte configuração:

Número de VAGs	1
Número de Pallets	8
Tamanho do Buffer	2
Regra para Despacho de Veículos	FIFO
Regra para Despacho de Peças	MTP

Tabela 6.1: Solução inicial (parâmetros e regras) para o problema exemplo

No anexo 3, são apresentadas as telas do protótipo, através das quais o usuário define todos os parâmetros e regras operacionais, desde a solução inicial até as interações subsequentes.

6.2.2 Simulação e Análise dos Resultados da Solução Inicial

Uma vez definidos os parâmetros e políticas operacionais, o modelo do SFM pode ser simulado. Após a execução da simulação uma mensagem informa ao usuário que o processo está terminado e que os resultados podem ser atualizados, vistos e analisados.

Utilizando a estrutura do Módulo de Análise vista no capítulo V, podemos verificar os resultados referentes as peças produzidas, as máquinas, os equipamentos, etc.. Também no anexo 3 podem ser observados os resultados das simulações, como eles se apresentam ao usuário nas telas e janelas do protótipo. As tabelas 6.2 a 6.5, apresentam os resultados da simulação do sistema exemplo, para os parâmetros da solução inicial.

	Peça 2	Peça 4	Peça 13	Peça 17	Peça 18
Produção	24	18	12	24	30
TMedFluxo	290,98	267,22	228,06	274,29	289,52

Tabela 6.2: Resultados da simulação inicial, referentes as peças

	Máquina 1	Máquina 2	Máquina 3	Máquina 4
Taxa Utilização	0,49	0,43	0,36	0,42
Tam. Méd. Fila	0,09	0,04	0,04	0,11

Tabela 6.3: Resultados simulação inicial, referentes as máquinas

Tamanho Médio da Fila de Peças Aguardando Transporte	4,848
Taxa média de Utilização	0,998

Tabela 6.4: Resultados simulação inicial, referentes aos VAGs

Tempo Médio do Ciclo de Produção	3782,228
Tempo Médio do Fluxo de Peças	275,916
Total de Peças Produzidas	108

Tabela 6.5: Resultados simulação inicial, referentes ao Sistema

Verifica-se pelos resultados apresentados que as metas de produção foram alcançadas com uma taxa de utilização máxima dos centros de usinagem igual a 49%, o que é considerado muito baixo para os padrões de SFM.

Por outro lado, observa-se que a utilização média dos VAGs foi de praticamente 100%

(99,8%). Esta alta taxa de uso, reflete a escassez de transportadores a qual pode ser verificada também pela alta média de peças na fila aguardando VAGs (4,85). A combinação destes resultados, mostra uma sub-utilização do sistema, em função do “gargalo” provocado pelo sistema de transportes.

Nesta primeira interação, os resultados das principais medidas de desempenho, isto é, o Tempo Médio do Ciclo de Produção (TMCP) e o Tempo Médio de Fluxo (TMFluxo) foram de 3782 min. e de 276 min., respectivamente.

É de se supor que um analista, observando os resultados e utilizando-se de seu bom senso e experiência, realizaria um novo experimento com um aumento no número de veículos na tentativa de solucionar o problema do gargalo.

Uma vez observados os resultados, o analista pode, se desejar, solicitar ao sistema uma análise e sugestões para a melhoria do desempenho operacional do SFM, via MA. Os resultados destes procedimentos são descritos e apresentados a seguir.

Para o exemplo que iniciamos, os resultados após a primeira rodada de simulação mostram que além das máquinas, com um diagnóstico de “Baixa Utilização”, o MA diagnosticou a utilização dos VAGs como “Alta Taxa de Utilização” e os tempos médios de fluxo ou de passagem das peças pelo sistema (TMFluxo) como “Alto”.

Diante destes resultados, as análises feitas pelas regras de otimização do sistema (ROTMS) indicam a possibilidade de melhorias no desempenho do sistema, uma vez que pode ser alterado o número de VAGs. Sua alta taxa de utilização, deve-se, principalmente a seu número reduzido.

Neste ponto, o MA sugere que se volte a simular com um incremento no número de VAGs. A figura 6.2, mostra uma tela da interface do protótipo com os resultados finais das análises feitas pelo MA.

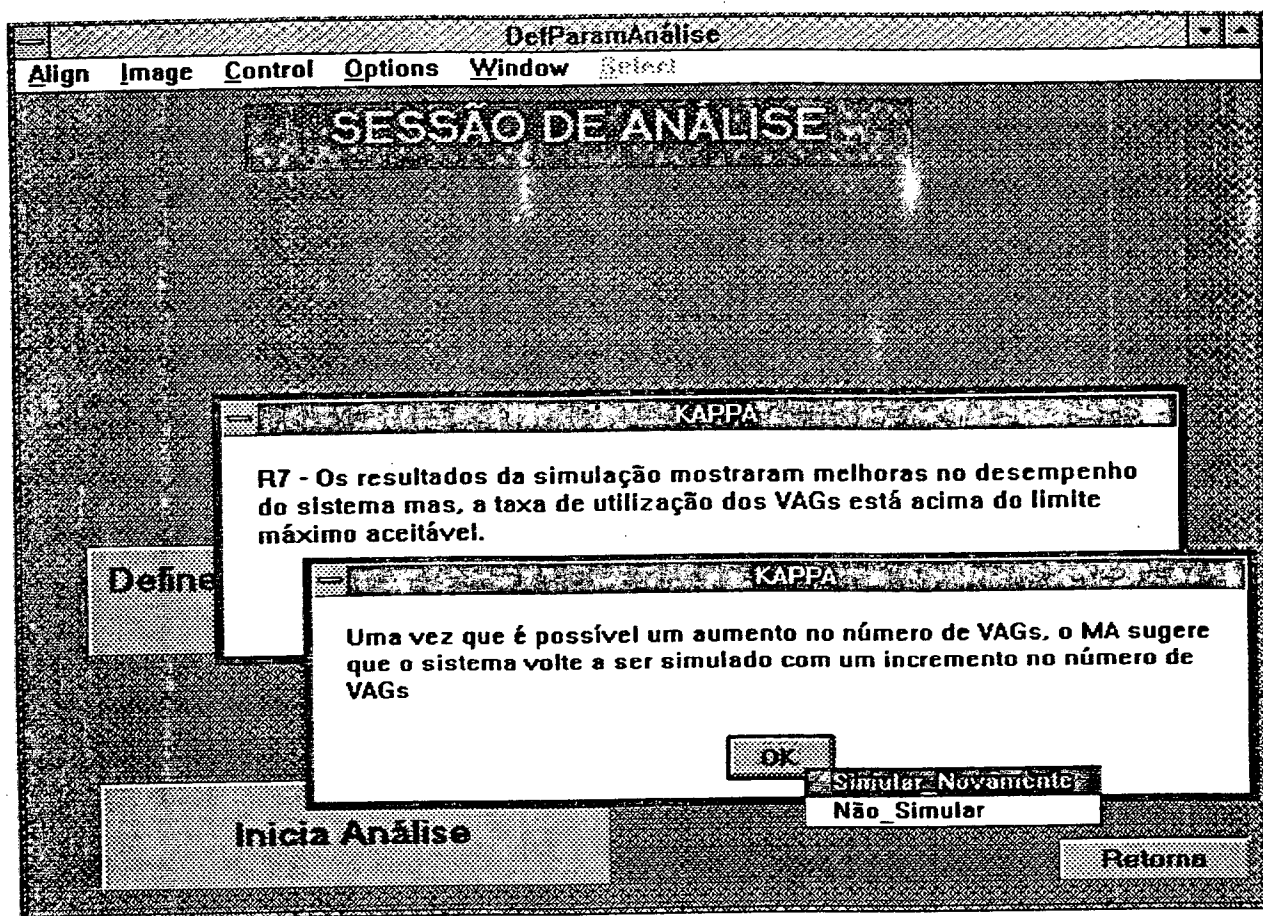


Fig. 6.2: Resultados das análises realizadas pelo MA para a simulação da solução inicial

6.2.3 Simulação e Análise dos Resultados das Interações Subsequentes

Nesta seção, apresentamos os resultados das demais interações entre o Módulo de Simulação e o analista, via Módulo de Análise.

Em cada uma delas, foram executadas modificações nos parâmetros operacionais antes do sistema ser novamente simulado. As modificações efetuadas e, aqui apresentadas, foram todas sugeridas pelo MA. Para cada interação executada, apresentamos uma tabela com seus principais resultados. Após a tabela, comentamos os resultados. Seguem-se as análises realizadas pelo MA, com uma figura ilustrativa da forma pela qual o sistema se comunica com o usuário/analista.

Interação 2:

Para a execução da segunda rodada de simulação, a única alteração efetuada no modelo foi sobre o número de VAGs, que passou de 1 para 2. Os demais parâmetros permaneceram os mesmos. Os principais resultados desta simulação indicam melhoras substanciais no desempenho do sistema e estão apresentados na tabela 6.6.

Interação	VAGs	Buffers	Pallets	TCP	TMFluxo	Util.Máx.Máq	Util.Méd.VAGs
2	2	2	8	2166	156	79%	96.8%

Tabela 6.6: Resultados da segunda simulação

Observamos na tabela as seguintes alterações sobre os resultados da primeira interação:

- houve uma redução em cerca de 42% (1616 min) no tempo médio do ciclo de produção;
- houve uma redução, sobre o tempo médio de fluxo das peças pelo sistema, de 276 min. da solução inicial, para cerca de 156 min, nesta simulação;
- a taxa de utilização dos centros de usinagem cresceu de 49% para 79%.
- a taxa de utilização dos VAGs, embora tenha se reduzido, ainda continua alta, com cerca de 97%
-

Os resultados mostram que, embora tenham havido muitas melhorias no desempenho geral do SFM, o problema anteriormente detectado continua, isto é, a alta taxa de utilização do VAGs.

Após a visualização dos resultados, o MA executa as análises. As mesmas regras são invocadas e o sistema propõe, novamente, que se aumente o número de VAGs na busca de um melhor desempenho, uma vez que o transporte interno à célula continua sendo o gargalo.

Uma vez aceita a sugestão, a alteração é automaticamente efetuada e o modelo simulado novamente. A forma de apresentação das análises pela MA é semelhante à apresentada na análise da simulação da solução inicial (ver figura 6.2), uma vez que o problema

detectado foi semelhante e, como consequência, as mesmas regras foram invocadas pela máquina de inferência para seu diagnóstico.

Interação 3:

Para a execução da terceira rodada de simulação, a única alteração efetuada no modelo foi sobre o número de VAGs, que passou de 2 para 3. Os demais parâmetros permaneceram os mesmos.

Os resultados indicam que a alteração efetuada se reflete positivamente na operacionalidade do sistema. Novamente obtém-se ganhos no tempo médio do ciclo de produção, sua principal medida de desempenho. A tabela 6.7 mostra os resultados.

Interação	VAGs	Buffers	Pallets	TCP	TMFluxo	Util.Máx.Máq	Util.Méd.VAGs
3	3	2	8	1956	141	88%	87%

Tab. 6.7: Resultados da terceira simulação

Observamos na tabela as seguintes alterações sobre os resultados da segunda interação:

- O lote todo é terminado em 1956 min. A redução neste tempo foi de aproximadamente 10% (210 min);
- As taxas de utilização das máquinas e VAGs foram de 88% e 87%, respectivamente. Estas taxas encontram-se abaixo do limite máximo estabelecido pelo analista (90%);
- O tempo de fluxo das peças também diminui, passando de 156 min. para 141 min.

Uma vez que o sistema alcançou uma taxa de utilização aceitável para os VAGs, o MA passa a experimentar alterações no número de pallets e no tamanho dos buffers. Ativado, o SE, baseando-se em regras que refletem nossa experiência, adquirida, principalmente, nas inúmeras simulações anteriormente efetuadas, indica que a melhor relação entre pallets/buffers/VAGs, encontra-se abaixo de seu limite.

O MA sugere então (ver figura 6.3), que se volte a simular com um incremento no número de pallets.

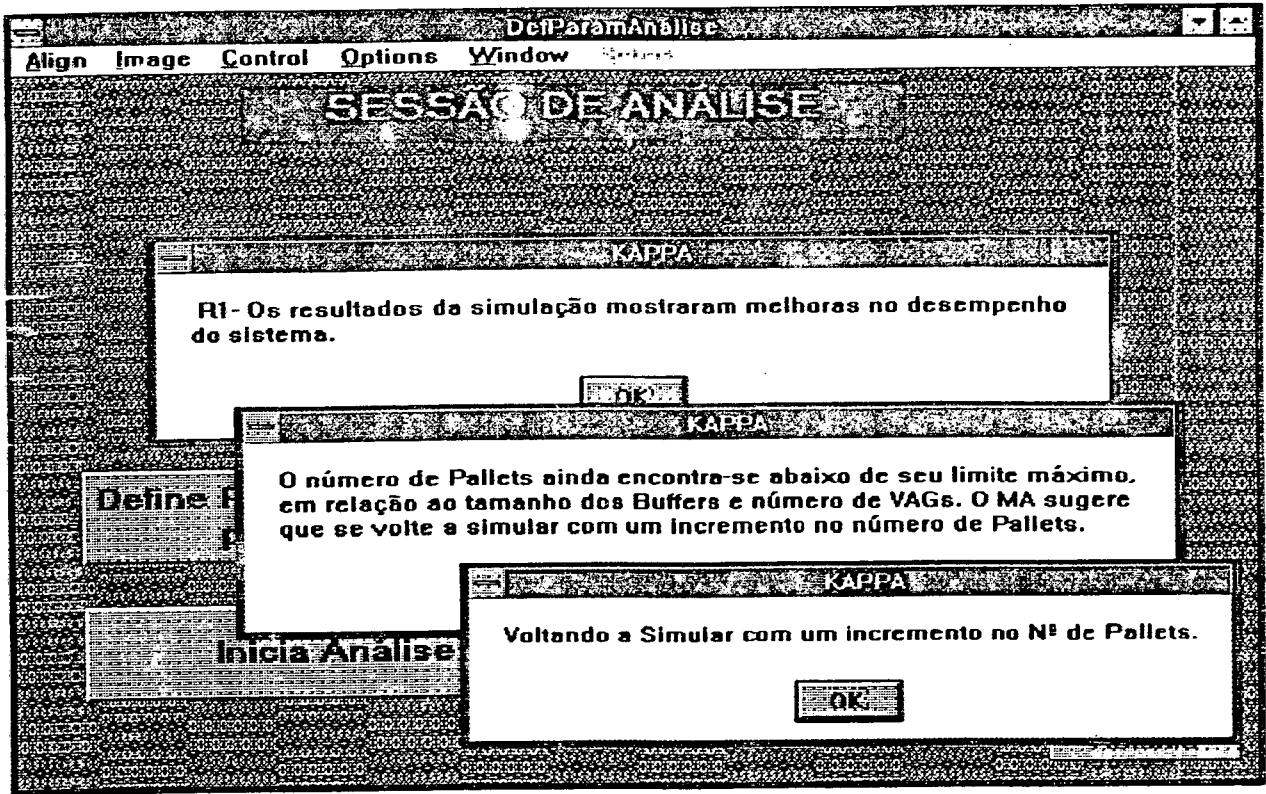


Figura 6.3: Tela com os resultados das análises da interação 3, elaboradas pelo MA

Interação 4:

O modelo volta a ser simulado com um incremento no número de pallets. Com este incremento, os pallets passam de 8 para 10. Os demais parâmetros permanecem os mesmos.

Interação	VAGs	Buffers	Pallets	TCP	TMFluxo	Util.Máx.Máq	Util.Méd.VAGs
4	3	2	10	1859	166	91%	91%

Tabela 6.8: Resultados da simulação 4

Podemos observar os seguintes resultados na tabela 6.8:

- Uma pequena melhora no tempo médio do ciclo de produção (de menos de 5%).
- Um aumento de 15% no tempo de fluxo das peças. Este aumento foi, possivelmente, causado pelo maior congestionamento no sistema, devido ao aumento do número de pallets.
- Um aumento nas taxas de utilização das máquinas e dos transportadores, possivelmente provocados pelo maior número de peças no sistema. As taxas voltam a superar os seus limites máximos.

Em resumo, não houve ganhos de performance no sistema. Com o crescimento do número de pallets, houve um provável desequilíbrio no sistema. Os transportadores ficam muitas vezes carregados esperando lugar nos buffers de entrada das máquinas, bloqueando temporariamente o sistema de transportes..

O MA sugere (ver figura 6.4) que incrementemos o tamanho dos buffers, visando um maior equilíbrio entre pallets/buffers/VAGs.

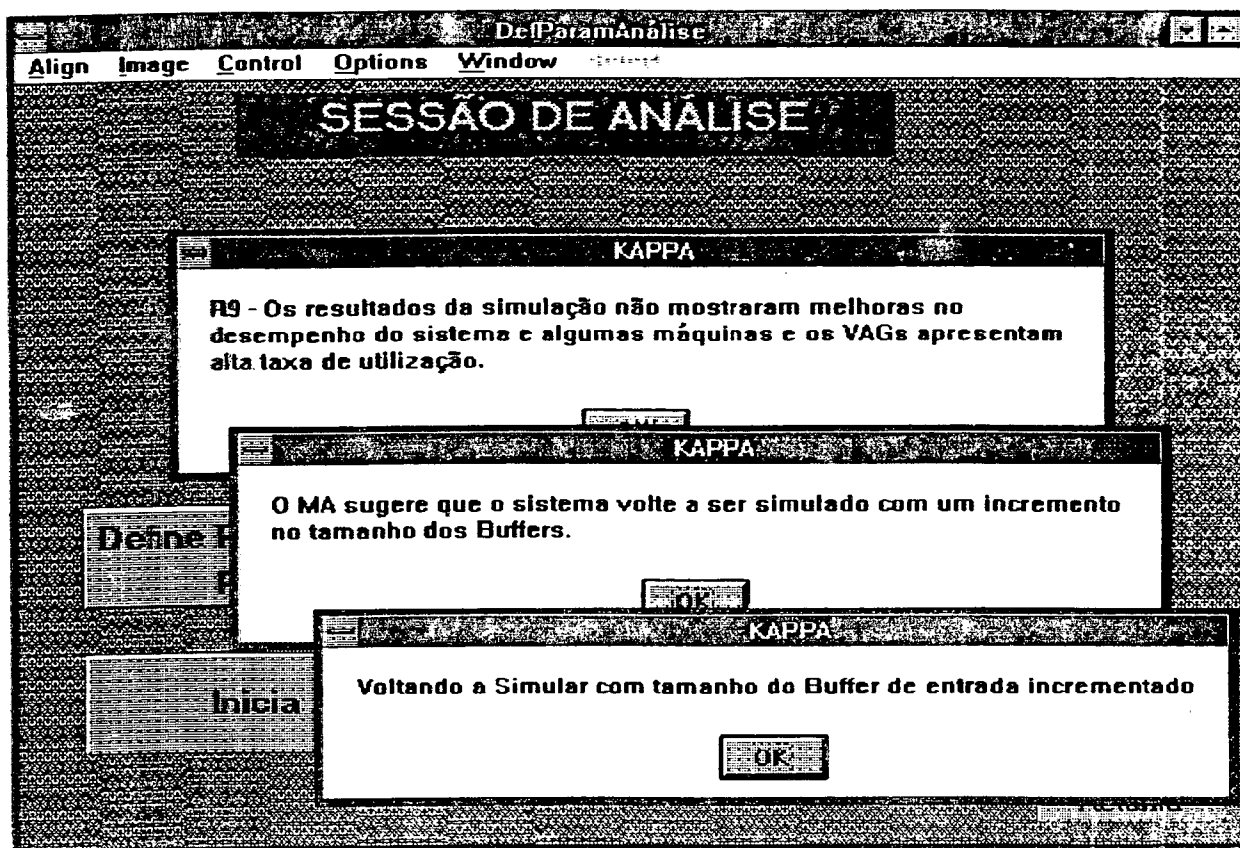


Figura 6.4: Tela com os resultados das análises da interação 4, elaboradas pelo MA

Interação 5:

O modelo volta a ser simulado com os buffers passando de tamanho 2 para 3. Os demais parâmetros permanecem os mesmos da interação anterior.

Os resultados da simulação são mostrados na tabela 6.9.

Interação	VAGs	Buffers	Pallets	TCP	TMFluxo	Util.Máx.Máq	Util.Méd.VAGs
5	3	3	10	1849	166	92%	88%

Tabela 6.9: Resultados da simulação 5

Mesmo com um maior número de peças no sistema, decorrentes do maior número de pallets, não ocorrem mais os bloqueios dos VAGs pela falta de espaço nos buffers de entrada dos centros de usinagem. Como se esperava, o aumento no tamanho dos buffers permite uma maior liberação dos VAGs, e um desafoço no fluxo de peças no sistema. Os principais resultados decorrentes disso são:

- A taxa de utilização dos VAGs cai para 88%.
- O tempo de fluxo não se altera, mostrando sua independência do número de buffers.
- O aumento no tamanho dos buffers provoca, no entanto, um pequeno aumento na taxa de utilização das máquinas e uma melhora quase insignificante no tempo de fabricação do lote.

Uma vez que este caminho mostra-se promissor, é possível crer que uma melhor combinação entre o tamanho dos buffers e o número de pallets, venha provocar uma redução na taxa de utilização das máquinas e no tempo do ciclo produtivo.

O MA sugere, como se pode ver na figura 6.5, que reproduz a tela do protótipo reportando o resultado das análises, que reduzamos o número de pallets e aumentemos o tamanho dos buffers, simultaneamente.

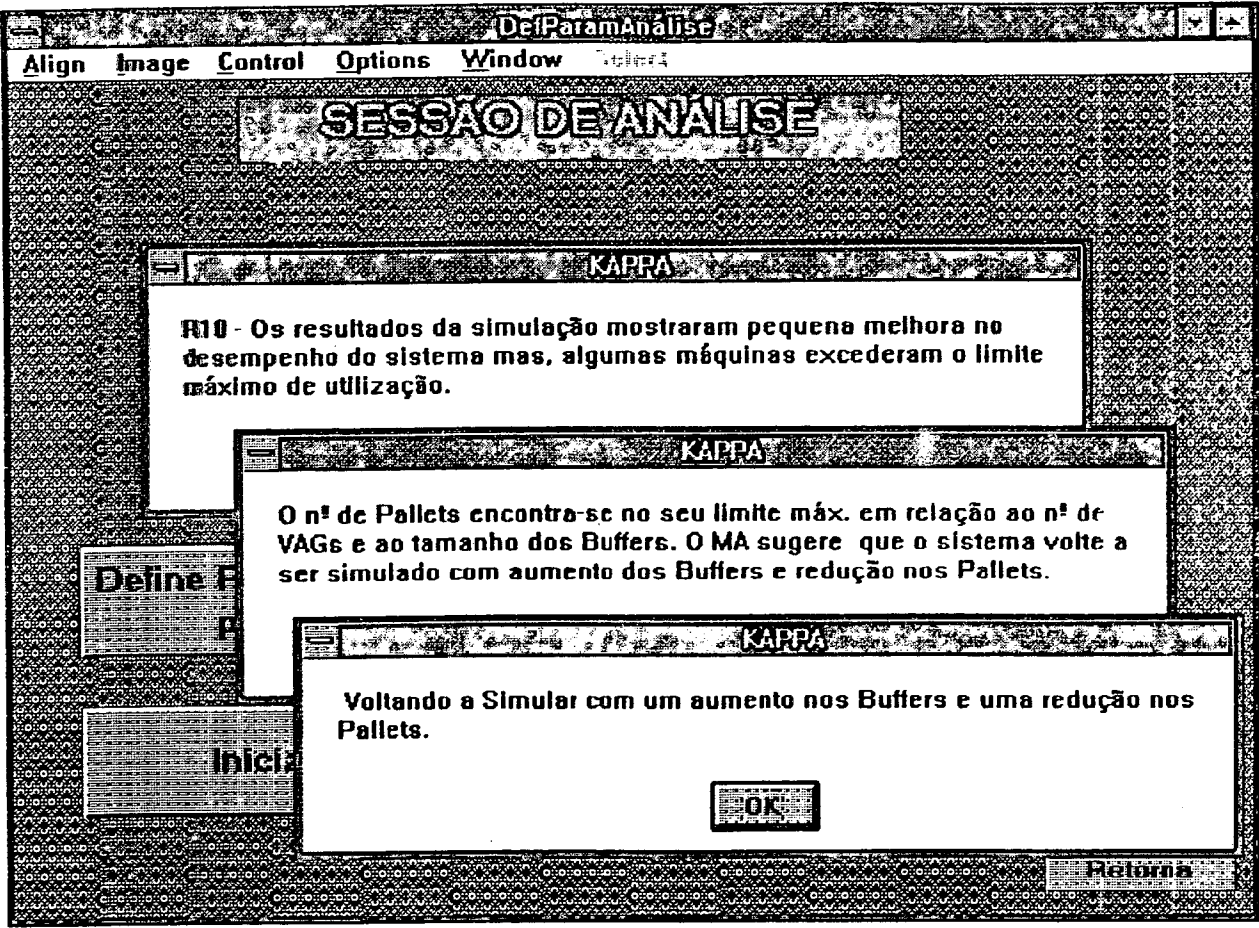


Figura 6.5: Tela com os resultados das análises da interação 5, elaboradas pelo MA

Interação 6:

A sexta simulação é realizada com tamanho dos buffers incrementados de 3 para 4 e o numero de pallets reduzidos de 10 para 8. conforme sugestão do MA. Os resultados da simulação podem ser vistos na tabela 6.10.

Interação	VAGs	Buffers	Pallets	TCP	TMFluxo	Util.Máx.Máq	Util.Méd.VAGs
6	3	4	8	1969	142	87%	85%

Tab. 6.10: Resultados da Interação 6

Os principais indicadores, apontam para um melhor balanceamento do sistema. Os resultados mais significantes são:

- As taxas de utilização das máquinas e VAGs caem para 87% e 85% respectivamente, abaixo do máximo estabelecido. Estes resultados são um pouco melhor do que os obtidos na 3ª interação, quando o sistema teve seu melhor desempenho.
- Os tempos de fluxo e de ciclo permanecem, também, no mesmo patamar daquela interação.

Os ganhos obtidos sobre os resultados da simulação anterior (quinta) são significativos, principalmente, considerando-se os limites estabelecidos para a utilização das máquinas e VAGs. No entanto, comparando-se estes resultados com aqueles obtidos na terceira simulação, estes ganhos não são substanciais.

A principal diferença nos parâmetros do sistema entre esta simulação e aquela, encontra-se no tamanho dos buffers, 2 e 4, respectivamente.

O desempenho do sistema mostrou-se muito dependente da relação número de pallets, tamanho dos buffers e número de VAGs. Na última simulação, com a redução no número de pallets, de 10 para 8, a relação entre os parâmetros relacionados aos buffers, pallets e VAGs, alcançam, novamente, um ponto próximo ao ideal, como já ocorrera na terceira simulação.

Um analista experimentado diria, provavelmente, que a solução está próxima do melhor desempenho deste sistema para a fabricação do lote de peças especificado.

As regras contidas no MA refletem este sentimento. Entre outras cláusulas embutidas nas regras, existe uma que estabelece qual deve ser o ganho mínimo sobre os resultados de simulações anteriores, que as modificações efetuadas sobre os parâmetros do sistema, devem provocar. Para o modelo utilizado, este mínimo ficou estabelecido em 5%. Como os ganhos obtidos não são substanciais, e as taxas de uso dos equipamentos encontram-se dentro dos limites estabelecidos, uma mensagem de encerramento do processo de Simulação/Análise é emitida, sugerindo a adoção para o sistema dos parâmetros da última simulação. A figura 6.6 reproduz a janela com a mensagem emitida pelo protótipo.

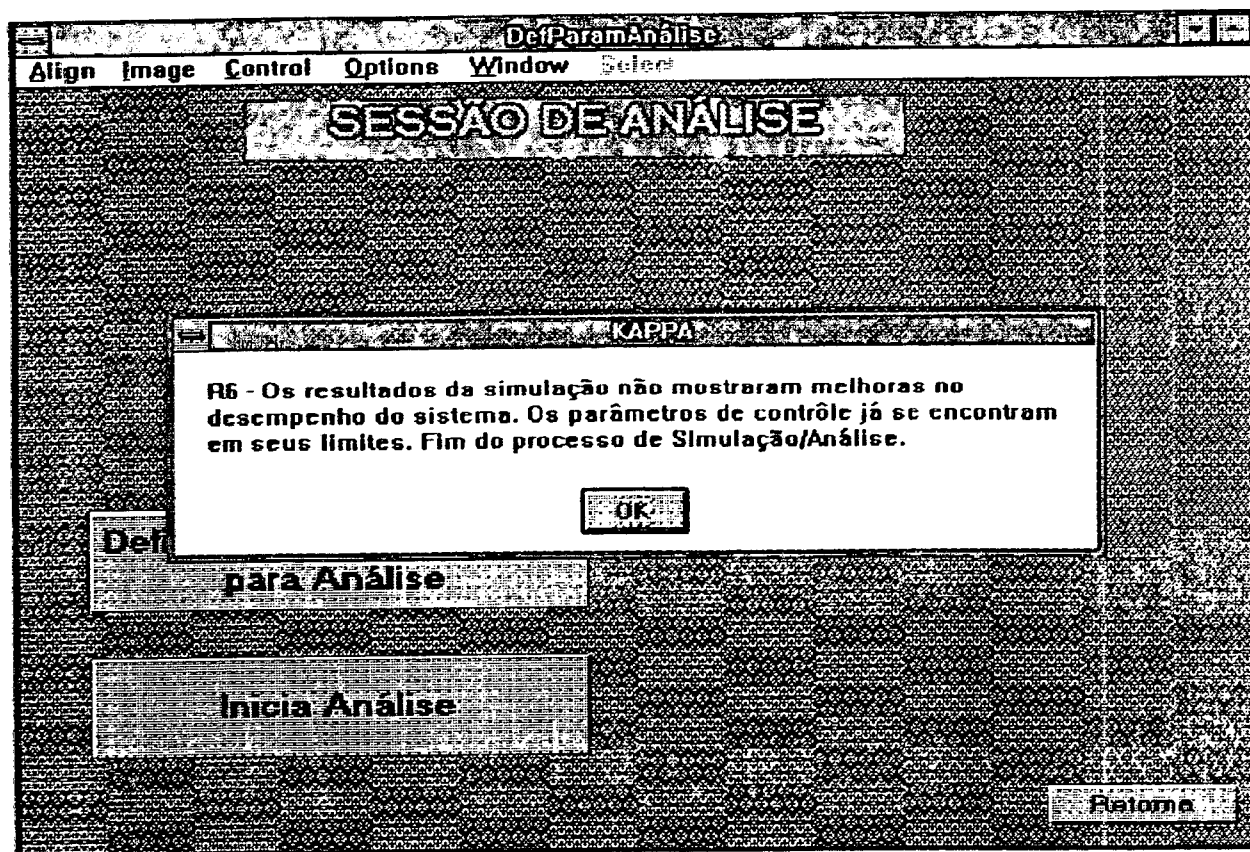


Figura 6.6: Tela com os resultados das análises da interação 6, elaboradas pelo MA

6.3 Comentários sobre os Resultados, Procedimento Interativo e Uso do Protótipo

Com base nas experiências realizadas, algumas observações podem ser apresentadas com respeito aos resultados, ao procedimento interativo e ao uso do protótipo

6.3.1 Comentários sobre os Resultados

1. Fica evidente, a partir dos resultados, a alta sensibilidade dos SFM às modificações em seus parâmetros operacionais, principalmente quando aqueles são expressos pelas taxas de utilização das máquinas e VAGs e pelos tempos médios de fluxo e de ciclo de produção.

Um exemplo muito claro disso ficou evidenciado pela dependência, destas medidas de desempenho, da relação estabelecida entre o número de pallets.

tamanho dos buffers e número de VAGs. Fora de uma relação próxima à ideal, a qual depende de cada sistema, em particular, as alterações em algum destes parâmetros podem provocar diferenças significativas no desempenho do sistema.

2. A taxa de utilização dos veículos é particularmente alta, principalmente pelo fato de ocorrerem um grande número de viagens sem carga, devido ao layout do sistema proposto, que admite somente o fluxo unidirecional dos VAGs. Estudos do layout e de otimização das viagens dos VAGs podem ser realizados utilizando-se o protótipo.
3. Os índices máximos de utilização das máquinas e VAGs, definidos pelo analista ao início dos procedimentos, é um fator importante, na medida em que índices muito altos podem ser operacionalmente suportados pelo sistema. Nos modelos tradicionais, em que planejamento e operação são considerados em momentos diferentes, estes fatores podem ser tomados como restrições no modelo de planejamento, em uma tentativa de viabilizar a operação. Em nosso modelo, em que a alocação das máquinas às peças é feita dinamicamente, é preciso tomar-se o cuidado de considerar tais restrições nos índices de eficiência das máquinas e nos valores de mínima e máxima utilização permitida para as máquinas e transportadores.

Nos exemplos similares simulados por MAHESHWARI [1992] e GUPTA [1992], nos quais o planejamento e a operação são considerados em momentos diferentes, os índices máximos de utilização das máquinas ficam em torno de 80%.

4. A otimalidade global dos resultados não está garantida pelo procedimento adotado. As soluções encontradas estão baseadas em uma pesquisa por pontos no espaço das soluções. Cada interação procura avaliar algum destes pontos. As análises posteriores, com base nos pontos já pesquisados, na experiência adquirida e armazenada na base de conhecimentos do SE e no bom senso, procuram direcionar a pesquisa para algum outro ponto dentro do espaço de soluções.
5. O número de problemas testados não é suficiente para concluir nada a respeito da taxa de convergência do procedimento. Nos casos testados, a convergência ocorreu em até sete interações, dependendo das restrições impostas pelo usuário.

6.3.2 Comentários sobre o Procedimento Interativo

1. O procedimento interativo requer, por parte do decisor, alguma sensibilidade e conhecimento sobre o comportamento operacional do SFM, diante das modificações, por ele efetuadas, sobre os parâmetros operacionais
2. A definição dos parâmetros iniciais é o principal ponto em que esta sensibilidade é mais requerida. A convergência poderá ser mais ou menos rápida, dependendo dos parâmetros iniciais e de suas futuras modificações.
3. A utilização das análises e sugestões feitas pelo MA, poderá, ou não, servir de guia para o usuário executar modificações nos parâmetros durante o procedimento interativo.
4. As modificações nos parâmetros estão sujeitas aos objetivos do usuário. Certos parâmetros produzem efeitos contrários sobre algumas medidas de desempenho. Como foi visto no capítulo IV, o número de pallets e tamanho dos buffers, por exemplo, causam reações opostas sobre o tempo de ciclo e sobre o tempo de fluxo.

6.3.3 Comentários sobre o Uso do Protótipo

1. O protótipo foi construído com dois propósitos: facilitar a experimentação sobre o modelo de simulação de SFM e permitir o uso do MA como auxiliar nas análises e tomadas de decisão.
2. A facilidade de conduzir uma experimentação é obtida em função da interface gráfica. Esta, permite ao usuário conhecer os resultados da simulação sob diferentes aspectos e de forma completamente diferenciada dos resultados padronizados, impostos pela linguagem de simulação original.
3. Algumas funções embutidas no protótipo permitem que, após cada rodada de simulação, o usuário possa, por meio de gráficos e tabelas, comparar resultados anteriormente obtidos com os atuais. Pequenos diagnósticos locais são também automaticamente gerados e apresentados ao analista, junto com os resultados da simulação, antes mesmo deste iniciar as análises via o MA.

4. O uso do sistema especialista, embutido no módulo de análise, não garante os melhores resultados, apenas informa ao usuário uma direção promissora e faz sugestões em função dos objetivos por ele traçados.

6.4 Sumário

Neste capítulo demonstramos o uso e aplicação do protótipo na condução de experimentos que envolvam a utilização do modelo de simulação para planejamento e operação de SFM.

Através de exemplos, demonstramos que sua interface gráfica facilita a condução destes experimentos e a observação dos seus resultados. Demonstramos também, que seu Módulo de análise constitui-se de um bom auxiliar na condução das análises que devem ser feitas sobre os resultados e nas sugestões sobre que alterações devem ser efetuadas nos parâmetros operacionais do sistema objetivando melhorar seu desempenho operacional.

No capítulo que segue, apresentaremos as conclusões finais sobre este trabalho de pesquisa, assim como sugestões e recomendações à futuras pesquisas.

CAPÍTULO VII

CONCLUSÕES E RECOMENDAÇÕES À FUTURAS PESQUISAS

Neste trabalho pesquisamos e buscamos novas alternativas para a solução de problemas associados ao planejamento e controle operacional de sistemas flexíveis de manufatura.

Propusemos um modelo para o tratamento de alguns destes problemas, acompanhado de um protótipo de sistema computacional, que combina duas ferramentas: simulação e sistemas especialistas.

Ao longo do trabalho, uma série de experimentos foram desenvolvidos e realizados, visando demonstrar a aplicação, utilidade e eficiência do modelo e do protótipo, propostos, que procura integrar parte da fase de planejamento à fase de controle operacional dos SFM.

Este capítulo faz uma síntese de tudo o que foi explorado, estudado e descoberto, apresentando os resultados obtidos e as contribuições consideradas relevantes. As limitações do trabalho também são apresentadas, assim como algumas recomendações a futuras pesquisas.

7.1 Resultados e Contribuições da Pesquisa

Os SFM possuem um enorme potencial para o aumento da produtividade nos sistemas de produção, em especial na produção seriada por lotes.

Sua principal atração é a possibilidade de obter-se alta produtividade mesmo com a produção de pequenos lotes.

Em contrapartida, estes sistemas exigem grandes investimentos de capital e necessitam especial atenção em todas as suas fases de implantação, desde o projeto até a posta em marcha.

Mesmo em regime de produção, a tarefa de planejar e controlar a operação de um SFM é mais complexa que nos sistemas tradicionais. Decisões inconsistentes em seu planejamento e controle operacional, podem levar, por exemplo, a uma baixa utilização de seus centros de usinagem e/ou um uso ineficaz de seu sistema automatizado de transporte, com conseqüentes congestionamentos no sistema, elevando, por exemplo, o tempo necessário para a fabricação de um lote de peças.

Resultados deste tipo são extremamente indesejados, uma vez que, a decisão de implantar células de manufatura altamente automatizadas, como os SFM, objetivam, principalmente, aumentar a função estratégica do setor produtivo, cuja principal contribuição, neste caso, é a redução dos ciclos de produção e o aumento da flexibilidade. Desta maneira, quando o objetivo é aumentar a eficiência dos SFM, as decisões acerca de seu planejamento e controle operacional devem ser precisas e confiáveis.

Em nosso estudo, examinamos uma série de problemas associados ao planejamento e controle operacional de SFM. Verificamos que uma das principais causas destes problemas, constitui-se na desvinculação entre o que é proposto pelo planejamento, e o que pode ser de fato realizado no nível operacional.

Tradicionalmente, entre as propostas do planejamento, está um prévio atrelamento entre máquinas e peças, em função de suas seqüências operacionais. Estas, determinam, previamente, onde cada operação deve ser realizada. Este tipo de atrelamento, é típico dos processos convencionais, os quais pouco consideram a capacidade multioperacional dos SFM e, portanto, sua alta flexibilidade, principalmente de rotas operacionais.

Desta forma, na grande maioria das vezes, as propostas da fase de planejamento, não são plenamente executáveis no nível operacional, uma vez que, naquela fase, a dinâmica do sistema é desconsiderada.

Uma das principais contribuições deste trabalho, foi desenvolver um modelo de simulação, apresentado no capítulo III, que permite considerar uma parte da fase de

planejamento durante a própria operação do sistema. Com isso, pretendemos que parte dos problemas de compatibilidade entre as duas fases, seja tratada de forma mais realista, uma vez que as decisões de alocação de máquinas às peças passa a ser feita de forma dinâmica, considerando parte do estado do sistema (carga de trabalho, filas, capacidade e eficiência das máquinas), momentos antes da decisão ser tomada.

O algoritmo embutido no modelo desenvolvido, simula a existência de um controlador que, em um sistema real, verificaria o estado das máquinas, peças e operações necessárias, designando então, o destino de cada peça dentro do sistema de acordo com os resultados destas análises. Um controlador deste tipo é perfeitamente possível em um SFM, uma vez que todas, ou a grande maioria das operações nestes sistemas, são controladas por computadores.

No capítulo IV deste trabalho, realizamos uma série de experimentos com diferentes propósitos. Com o objetivo de comprovarmos a eficiência e aplicabilidade do modelo desenvolvido, fez-se um trabalho comparativo, no qual tratamos um determinado problema sob duas formas de abordagem: utilizando o MPF, e outro, por nós classificado de tradicional, o qual faz uso da programação linear para a fase de planejamento e da simulação para a fase operacional. O problema-teste, foi criado, tratado e documentado em um recente trabalho de pesquisa [MAHESHWARI, 1992].

Os resultados obtidos, foram significativos, obtendo-se, em favor MPF, melhorias no desempenho do SFM, que variaram de aproximadamente 6% a 22%, quando se utilizou dos Tempos de Ciclo de Produção (TCP), como variável de controle, e de 3% a 14%, quando a variável utilizada foi o Tempo Médio de Fluxo (TMFluxo) das peças pelo sistema. Observamos, que as maiores diferenças ocorreram quando o sistema, em função dos valores adotados para os inúmeros parâmetros operacionais, encontrava-se mais congestionado. Cremos, que a principal causa destes ganhos sobre a abordagem tradicional, está relacionada com a falta de flexibilidade desta, pela utilização da programação linear na fase de planejamento, desconsiderando a dinâmica do sistema que, como vimos, tem forte dependência dos valores atribuídos aos recursos secundários do sistema.

Ainda no capítulo IV realizamos uma série de outros experimentos, estudando o comportamento do sistema em função das diversas combinações de parâmetros e políticas operacionais.

Duas foram as razões destes novos experimentos. Primeiramente, procuramos verificar a sensibilidade do modelo em traduzir o impacto destas combinações sobre o desempenho operacional do sistema. Como vimos na revisão da literatura, alguns trabalhos reportam experimentos semelhantes, o que nos permitiu, mais uma vez, comprovar a validade do modelo, uma vez que, aplicando-se dados semelhantes obtivemos resultados que apontaram na mesma direção.

Comprovou-se, por exemplo, que o controle sobre a ordem de liberação das peças para o sistema, tem grande influência sobre as duas variáveis de controle (TCP e TMFluxo). Quando se utiliza uma política (regra) que procura manter balanceado o mix de produção, os resultados são superiores aqueles obtidos com outras regras. Outros resultados que confirmam aqueles obtidos por outros pesquisadores foram as influências exercidas pelo tamanho dos buffers e pelo número de pallets, sobre as mesmas variáveis. O aumento dos buffers, por exemplo, reduz o TCP e aumenta o TMFluxo. A decisão sobre este parâmetro, portanto, dependerá, fundamentalmente, dos objetivos definidos pelo decisor.

A segunda razão para esta nova série de experimentos, foi buscar uma maior compreensão dos fenômenos envolvidos no comportamento operacional destes complexos sistemas de produção, diante das inúmeras possibilidades associadas ao seu controle, uma vez que, um dos objetivos deste trabalho consistia na construção de um sistema especialista que pudesse auxiliar os tomadores de decisão na tarefa de planejar e controlar suas operações. Desta forma, parte da experiência e do conhecimento necessários, foram adquiridos nestes experimentos e, posteriormente, traduzidos e incorporados a um módulo inteligente de análise, na forma de regras de produção, frames, métodos, etc..

Todo o processo de planejamento e controle da produção de SFM, empregando ou não modelos matemáticos, não é tarefa simples e não costuma ser executada em um só passo. Ao contrário, este processo costuma ser longo e repetitivo, constituindo-se, na verdade, de uma série de experimentos efetuados pelo decisor, em que uma série de alternativas são testadas antes de sua real implementação. Evidentemente, dependendo do grau de

conhecimento, experiência e ferramentas auxiliares utilizadas pelo decisor, este processo poderá ser simplificado e abreviado.

Diante desta realidade, propusemos e concretizamos o desenvolvimento do protótipo de um sistema computacional que combina o modelo de simulação anteriormente apresentado, com um sistema especialista. Além disso, com o intuito facilitar, tanto seu uso como todo o processo decisório, dotamos o protótipo de uma interface gráfica voltada ao usuário.

Consideramos este protótipo a segunda maior contribuição deste trabalho uma vez que, através dele, procuramos encaminhar uma solução para o problema, sempre presente, das dificuldades associadas ao uso de modelos, particularmente os modelos de simulação. A interface possibilita, com facilidade, a execução de experimentos.

A combinação do SE com o modelo de simulação, permite que o usuário, na medida em que vai acumulando experiência, possa guardar, ao menos parte desta, sob a forma de regras de produção ou mesmo métodos, os quais, uma vez implantados na base de conhecimentos, podem vir a auxiliá-lo em experimentos futuros.

Neste protótipo, as duas metodologias interagem de forma paralela, uma vez que duas diferentes ferramentas (uma linguagem de simulação e um ambiente de desenvolvimento de sistemas inteligentes) foram utilizados na sua construção. Os resultados de uma influem no desempenho da outra.

7.2 Limitações do Trabalho

Como qualquer trabalho de pesquisa, a abrangência deste estudo é também limitada. Seus limites foram definidos e apresentados ao longo das várias etapas desenvolvidas. Todos os resultados numéricos foram obtidos sobre um sistema fictício, embora sua modelagem tenha sido, originalmente, baseada num sistema real. Desta forma, no seu sentido mais estrito, todas as conclusões derivadas dos experimentos são válidas apenas para aquele sistema, somente.

No que diz respeito ao modelo e ao sistema computacional desenvolvidos, estes são mais genéricos por sua natureza, aplicando-se a outros sistemas semelhantes, podendo, no

entanto, ser ampliados melhorados, explorando outros aspectos do problema ao qual se propõem tratar.

Os experimentos realizados, consideraram apenas algumas poucas estratégias de controle operacional. Existem muitas outras que podem ser exploradas sobre o SFM tratado ou sobre outros sistemas de manufatura semelhantes.

7.3 Recomendações para Futuras Pesquisas

As pesquisas realizadas neste trabalho, sugerem algumas áreas relevantes para o desenvolvimento outros estudos. Nesta seção apontamos para algumas destas e recomendamos futuros trabalhos de pesquisa. Os itens abordados não seguem nenhuma ordem de prioridade.

7.3.1 Extensão da Capacidade do Módulo de Análise

Uma das áreas mais promissoras ao desenvolvimento de futuros trabalhos de pesquisa, seguramente é o desenvolvimento da capacidade de diagnóstico, análise e tomada de decisão do Módulo de Análise (MA), incorporado ao protótipo. Em seu atual estágio, este módulo suporta sistemas flexíveis de manufatura com características semelhantes ao sistema tratado. Serão valiosas, contribuições que permitam ao MA suportar diferentes características físicas associadas aos SFM, como por exemplo, um sistema de transporte de materiais caracterizado por esteiras no lugar de ou, em conjunto com, os VAGs.

Para que o sistema especialista possa suportar outras características físicas, sugerimos que sejam implementados conjuntos de módulos de análise, semelhantes ao já desenvolvido, específicos para diferentes características físicas de SFM. Uma vez disponíveis, antes de iniciar qualquer processo, o decisor poderia compor seu MA, incorporando este ou aquele módulo necessário aos diagnósticos ou análises do sistema de produção com o qual estivesse lidando.

Outro importante campo de pesquisas, é da incorporação novas regras ao sistema especialista que possam auxiliar o decisor na fase de planejamento da produção. Alguns elementos que podem ser implementados dizem respeito, por exemplo, a definição do mix e do volume de produção das peças (tal decisão pode estar atrelada, por exemplo, as datas

de entrega dos lotes de produzidos), a definição dos fatores de utilização dos recursos, a capacidade do sistema de transporte, etc.

Outra área de pesquisa desejável, ainda sobre o módulo inteligente, diz respeito a forma e facilidade de incorporação do conhecimento ao sistema especialista, via novas regras. É desejável que se possa automatizar estes procedimentos. Ao mesmo tempo, é desejável que se criem procedimentos, tais como metaregras, que procurem manter a consistência global da base de conhecimentos e da máquina de inferência.

7.3.2 Extensão da Capacidade do Módulo de Simulação

Outra grande área atuação para o desenvolvimento de novos estudos relacionados a este trabalho, diz respeito a capacidade do Módulo de Simulação (MS). Quando construímos este módulo, tínhamos em mente aproveitar ao máximo a capacidade da linguagem de simulação (SIMAN) na qual modelamos o SFM exemplo. De fato, cremos ter alcançado este objetivo. Contudo, com o aumento da capacidade e potencialidade das linguagens de simulação, será possível, muito em breve, incorporar outras características ao MS.

Na medida em que disponhamos de um ambiente de simulação orientado a objeto, e que seja executável na mesma plataforma do protótipo (Windows), será potencialmente possível ampliar sobremaneira e, implementar dentro do MS, a capacidade de modelar ou modificar integralmente modelos de sistemas de produção. No atual estágio do MS, tal capacidade está restringida a alteração de parâmetros e valores de variáveis ou atributos do modelo desenvolvido.

Outra capacidade que pode ser incorporada ao MS, é a animação dos modelos simulados. Ao tempo do desenvolvimento do trabalho, não tínhamos a disposição as ferramentas de animação para a versão da linguagem SIMAN com a qual trabalhamos. Hoje estas ferramentas já foram adquiridas, e podem vir a incorporar os modelos de simulação desenvolvidos para o MS.

7.3.3 Integração do Fluxo de Ferramentas ao Modelo.

No modelo de simulação desenvolvido, consideramos que os diversos centros de usinagem possuem, ao início de cada ciclo de produção, um determinado conjunto de

ferramentas em seu magazine, necessários a execução de determinadas operações. É possível, no entanto, considerar a existência em SFM, de transportadores e manuzeadores automáticos de ferramentas, os quais poderiam movimenta-las entre os centros, proporcionando uma maior flexibilidade ao sistema. Desta forma, se considerarmos esta possibilidade, as restrições sobre que operações os centros de usinagem podem realizar, poderão ser menos limitadas, aumentando a potencial flexibilidade do sistema.

7.3.4 Outras Extensões

Algumas outras idéias associadas a futuras pesquisas, são listadas abaixo:

- Implementação de regras que analisem o impacto de diferentes layouts do sistema.
- Implementação de outros algoritmos, junto ao modelo de simulação, que busquem outros objetivos, tais como, minimização de custos operacionais, minimização da soma dos tempos de processamento e transporte, etc
- Implementação de um módulo para análise de um índice de flexibilidade alcançado pelo sistema, em função de sua flexibilidade potencial e utilizada. Um módulo deste tipo, pode ser extremamente útil quando da necessidade de comparar-se distintos sistemas para fins, por exemplo, de decisões estratégicas.

ANEXOS

ANEXO 1

CÓDIGO FONTE DO PROGRAMA PARA ALOCAÇÃO DINÂMICA DOS
RECURSOS EM TEMPO DE SIMULAÇÃO

```

/*****
*
* Module:   USERC.C
*
*****
*/

#include "simlib.h"

#ifdef SUN_

# define cprime  cprime_
# define cwrap   cwrap_
# define cevent  cevent_
# define cuf     cuf_
# define cur     cur_
# define cstate  cstate_
# define cuclea  cuclea_
# define cusav   cusav_
# define curst   curst_
# define ckeyhi  ckeyhi_
# define cuerr   cuerr_
#endif

#ifdef _MSC_VER

# define ms_float double
#else
# define ms_float float
#endif

```

```

/*****
*
*
* Routine :   cprime
*
*****/

*/
static float matriz[20][8][2]; /** x=pecas, y=operacoes e z=0->tipo oper**/
static float efic[4][10]; /** x=maq, y=oper**/ /** z=1->tempo oper**/
static float EntBuf,SaiBuf;
static smint lib[5];
static smint mix[100];

#include "c:\c700\include\stdio.h"
#include "c:\c700\include\stdlib.h"
#include "c:\c700\include\ctype.h"

FILE *fp;

void cprime ( sim )
simstr *sim;

{
smint i;
smint y;
smint z;
char string[11];

/**Busca Dados do KAPPA**/
/*****
*/
/** x(0)=EntBuf, x(1)=SaiBuf, x(2)=NumPall, x(3)=NumOper, x(4)=NumPeca
**/
/** x(5)=NumMaq, x(6)=NMaxEnt ****/
/*****
*/

fp=fopen("dadkap13.txt","rt");

```

```

for(i=0;i<=6;i++)          /** Buffer de saida e entrada **/
{
    /** NumPall, NumOper, NumPeca, NumMaq **/
    fgets(string,10,fp);
    sim->x[i]=atof(string);
}
for(i=0;i<=(sim->x[4]);i++) /** Mix das pecas **/
{
    fgets(string,10,fp);
    mix[i]=atof(string);
    printf("\nmix[i]%f", (float)mix[i]);
}

/** Busca Caracteristicas do sistema **/
/** Peca X NumOper (i=NumPeca y=NumOp)**/

for(z=0;z<=1;z++)          /** z=0 ->tipo op., z=1 -> tempo op.**/
{
    for(i=0;i<=(sim->x[4]);i++)
    {
        for(y=0;y<=(sim->x[3]);y++)
        {
            fgets(string,10,fp);
            matriz[i][y][z]=atof(string);
        }
    }
}

/** maquina X Operacao **/      /** i=maquina y=tipo de operacao**/

for(i=0;i<=(sim->x[5]);i++)
{
    for(y=0;y<=(sim->x[3]);y++)
    {
        fgets(string,10,fp);
        efic[i][y]=atof(string);
    }
}
fclose(fp);

/** getchar();**/
return;
}

```

```

/*****
*
*
* Routine :   cwrap
*
*****/

void cwrap ( sim )

simstr *sim;
{
fp=fopen("T13out.001","wt");
fclose(fp);
return;
}
/*****
*
* Routine :   cevent
*
*****/

void cevent ( Lent, Nevt, sim )

smint *Lent;
smint *Nevt;
simstr *sim;
{

register smint peca;  /** trabalhar como atributo **/
register smint operacao;  /** trabalhar como atributo **/
register smint maquina;
register smint tipo;
static float TDispo[10];

peca=(smint)sim->d[0];
operacao=(smint)sim->d[1];

```



```
maquina=(smint)sim->d[4];
```

```
peca--;
```

```
operacao--;
```

```
maquina--;
```

```
tipo=(smint)(matriz[peca][operacao][0]);
```

```
tipo--;
```

```
/****** EVENT 1 *****/
```

```
if ((int)*Nevt==1)
```

```
{
```

```
    float NovaCarga,CargaTeste;
```

```
    smint MaqEsc,m=0;
```

```
    for(m=0;m<=(sim->x[5]);m++)
```

```
    {
```

```
        if ((float)TDispo[m]<=(float)sim->tnow)
```

```
        {
```

```
            TDispo[m]=sim->tnow;
```

```
        }
```

```
    }
```

```
    if((float)tipo!=-1)
```

```
    {
```

```
        m=0;
```

```
        while((float)efic[m][tipo]==0)** || sim->nq(&m)>MaxBuf)**
```

```
        {
```

```
            m++;
```

```
        }
```

```
        MaqEsc=m;
```

```
        NovaCarga=TDispo[m]+matriz[peca][operacao][1]
```

```
            /efic[m][tipo];
```

```
        for(m=++m;m<=(sim->x[5]);m++)
```

```
        {
```

```
            if((float)efic[m][tipo]!=0)** || sim->nq(&m)<=MaxBuf)**
```

```
            {
```

```
                CargaTeste=(TDispo[m]+matriz[peca][operacao][1]
```

```
                    /efic[m][tipo]);
```

```
                if (NovaCarga>(float)CargaTeste)
```

```
                {
```

```
                    printf("\n    O valor de m:%f", (float)m); **/
```

```
                    NovaCarga=CargaTeste;
```

```
                    MaqEsc=m;
```

```
                }
```

```
            }
```

```
        }
```

```

    }
  }
}
TDispo[MaqEsc]=(float)NovaCarga;
sim->d[2]=(float)(MaqEsc+1);
}
else if((float)tipo==1)
{
  sim->d[2]=6;
}
}

/***** EVENT 2 *****/
else if ((int)*Nevt==2)
{
  printf("\neficiencia :%f\n", (float)efic[maquina][tipo]);**/
  sim->d[3]=(float)matriz[peca][operacao][1]/(float)efic[maquina][tipo];
}

return;
}

/*****
*
*
* Routine :   cuf
*
*****/

ms_float cuf ( l, n, sim )

smint *l;
smint *n;
simstr *sim;

{
  smint i;
  float PEsc, xmin, xmax;
  xmin=(float)((float)lib[0]/(float)mix[0]);

```

```
PEsc=0;
for(i=1;i<=(sim->x[4]);i++)
{
    xmax=(float)((float)(lib[i])/(float)mix[i]);
    if(xmin>xmax)
    {
        xmin=(float)(lib[i]/mix[i]);
        PEsc=(float)i;
    }
}
lib[(smint)(PEsc)]++;
return((ms_float)(PEsc+1));
}
```

ANEXO 2

CÓDIGO FONTE COM TODAS REGRAS DO PROTÓTIPO

```

/*****
**      ALL RULES ARE SAVED BELOW      **
*****/

/*****
**** RULE: RDGS1
*****/
MakeRule( RDGS1, [],
  Sistema:TCPSim <= Sistema:TCPEst.
  Sistema:DiagTCP #="TCP Aceitável" );

/*****
**** RULE: RDGS2
*****/
MakeRule( RDGS2, [],
  Sistema:TCPSim > Sistema:TCPEst.
  Sistema:DiagTCP #="TCP Não Aceitável" );

/*****
**** RULE: RDGS3
*****/
MakeRule( RDGS3, [],
{
  ForAll [ maq|Máquinas ]
    maq:Diag_Util_Máq #= Normal Or maq:Diag_Util_Máq #=
      Baixa:
},
  Sistema:DiagUtilMáq = Aceitável );

/*****
**** RULE: RDGS4
*****/
MakeRule( RDGS4, [],
{
  ForAll [ maq|Máquinas ]
    maq:Diag_Util_Máq #= Normal Or maq:Diag_Util_Máq #=
      Baixa:
},
  Sistema:DiagUtilMáq = "Não Aceitável" );

/*****
**** RULE: RDGS5
*****/
MakeRule( RDGS5, [],
  VAGs:Util_Diag #= Baixa.

```

Sistema:DiagUtilVAGs = "Baixa Utilização");

```
/******
**** RULE: RDGS6
*****/
```

MakeRule(RDGS6, [],
VAGs:Util_Diag \neq Normal,
Sistema:DiagUtilVAGs = "Utilização Normal");

```
/******
**** RULE: RDGS7
*****/
```

MakeRule(RDGS7, [],
VAGs:Util_Diag \neq Alta,
Sistema:DiagUtilVAGs = "Alta Utilização");

```
/******
**** RULE: RDGS8
*****/
```

MakeRule(RDGS8, [],
Sistema:TFxPSim \leq Sistema:TMédFxPEst,
Sistema:DiagTMFP = "TMF Aceitável");

```
/******
**** RULE: RDGS9
*****/
```

MakeRule(RDGS9, [],
Sistema:TFxPSim \leq Sistema:TMédFxPEst,
Sistema:DiagTMFP = "TMF Não Aceitável");

```
/******
**** RULE: ROTm1
*****/
```

MakeRule(ROTm1, [],
LengthList(Máquinas:ListaMáqSobreUtil) \leq 0 And VAGs:UtilMédReal
 \leq VAGs:MaxUtilVAGs And Sistema:TMédCPSim $<$ Sistema:TMédCPSimAnt
* 0.95 And Máquinas:TamBufferEnt \leq Máquinas:TamMáxBufferEnt
And Pallets:NPallets \geq Pallets:NMinPallets And Pallets:NPallets
 $<$ Pallets:NMaxPallets.

{
Sistema:DiagSistema = " R1- Os resultados da simulação mostraram melhoras no desempenho do sistema.";

PostMessage(Sistema:DiagSistema);

PostMessage("O número de Pallets ainda encontra-se abaixo de seu limite máximo, em relação ao tamanho dos Buffers e número de VAGs. O MA sugere que se volte a simular com um incremento no número de Pallets. ");

Sistema:Menu = SimÑSim1;

} E

```
/******
**** RULE: ROTm2
*****/
```

MakeRule(ROTm2, [],
LengthList(Máquinas:ListaMáqSobreUtil) \leq 0 And VAGs:UtilMédReal

```

> VAGs:MaxUtilVAGs And Sistema:TMédCPSim < Sistema:TMédCPSimAnt
* 0.95 And Máquinas:TamBufferEnt <= Máquinas:TamMáxBufferEnt
And Pallets:NPallets > Pallets:NMinPallets And Pallets:NPallets
< Pallets:NMaxPallets,
{
Sistema:DiagSistema = "R2 - Os resultados da simulação mostraram melhoras no desempenho do
sistema mas, a Utilização Média dos VAGs está acima do máximo aceitável.";
PostMessage( Sistema:DiagSistema );
PostMessage( "Nº de Pallets no limite com relação ao nº de VAGs e aos Buffers. O MA sugere que se
volte a simular com um aumento nos Buffers e uma redução nos Pallets." );
Sistema:Menu = SimÑSim2;
} );

/*****
**** RULE: ROTm3
*****/
MakeRule( ROTm3, [],
LengthList( Máquinas:ListaMáqSobreUtil ) <= 0 And VAGs:UtilMédReal
<= VAGs:MaxUtilVAGs And Sistema:TMédCPSim < Sistema:TMédCPSimAnt
* 0.95 And Máquinas:TamBufferEnt <= Máquinas:TamMáxBufferEnt
And Pallets:NPallets #= Pallets:NLimPallets,
{
Sistema:DiagSistema = "R3 - Os resultados da simulação mostraram melhoras no desempenho do
sistema. ";
PostMessage( Sistema:DiagSistema );
Sistema:Menu = SimÑSim3;
} );

/*****
**** RULE: ROTm4
*****/
MakeRule( ROTm4, [],
LengthList( Máquinas:ListaMáqSobreUtil ) <= 0 And VAGs:UtilMédReal
<= VAGs:MaxUtilVAGs And Sistema:TMédCPSim >= Sistema:TMédCPSimAnt
* 0.95 And Máquinas:TamBufferEnt <= Máquinas:TamMáxBufferEnt
And VAGs:NumVAGs < VAGs:NumMaxVAGs,
{
Sistema:DiagSistema = "R4 - Os resultados da simulação não mostraram melhoras no desempenho do
sistema. ";
PostMessage( Sistema:DiagSistema );
PostMessage( "Alcançado o máximo desempenho do sistema com o atual nº de VAGs. O MA sugere
que o sistema volte a ser simulado com um incremento no número de VAGs." );
Sistema:Menu = SimÑSim4;
} );

/*****
**** RULE: ROTm5
*****/
MakeRule( ROTm5, [],
LengthList( Máquinas:ListaMáqSobreUtil ) > 0 And VAGs:UtilMédReal
<= VAGs:MaxUtilVAGs And Sistema:TMédCPSim < Sistema:TMédCPSimAnt
* 0.95 And Máquinas:TamBufferEnt <= Máquinas:TamMáxBufferEnt,
{

```

Sistema:DiagSistema = "R5 - Os resultados da simulação mostraram melhoras no desempenho do sistema mas, algumas máquinas excederam o limite máximo de utilização. ";

PostMessage(Sistema:DiagSistema);

Sistema:Menu = SimÑSim5;

});

/******

**** RULE: ROtm6

*****/

MakeRule(ROtm6, [],

LengthList(Máquinas:ListaMáqSobreUtil) <= 0 And VAGs:UtilMédReal

<= VAGs:MaxUtilVAGs And Sistema:TMédCPSim >= Sistema:TMédCPSimAnt

* 0.95 And VAGs:NumVAGs #= VAGs:NumMaxVAGs.

{

Sistema:DiagSistema = "R6 - Os resultados da simulação não mostraram melhoras no desempenho do sistema. Os parâmetros de controle já se encontram em seus limites. Fim do processo de Simulação/Análise. ";

PostMessage(Sistema:DiagSistema);

Sistema:Menu = Menu6;

});

/******

**** RULE: ROtm7

*****/

MakeRule(ROtm7, [],

LengthList(Máquinas:ListaMáqSobreUtil) <= 0 And VAGs:UtilMédReal

> VAGs:MaxUtilVAGs And (Sistema:TMédCPSim < Sistema:TMédCPSimAnt

* 0.95 Or Sistema:TMédCPSim

< Sistema:TCPEst)

And VAGs:NumVAGs < VAGs:NumMaxVAGs And Pallets:NPallets

#= Pallets:NMinPallets.

{

Sistema:DiagSistema = "R7 - Os resultados da simulação mostraram melhoras no desempenho do sistema mas, a taxa de utilização dos VAGs está acima do limite máximo aceitável. ";

PostMessage(Sistema:DiagSistema);

PostMessage("Uma vez que é possível um aumento no número de VAGs, o MA sugere que o sistema volte a ser simulado com um incremento no número de VAGs");

Sistema:Menu = SimÑSim7;

});

/******

**** RULE: ROtm8

*****/

MakeRule(ROtm8, [],

LengthList(Máquinas:ListaMáqSobreUtil) > 0 And VAGs:UtilMédReal

> VAGs:MaxUtilVAGs And Sistema:TMédCPSim <= Sistema:TMédCPSimAnt

* 0.95 And Máquinas:TamBufferEnt <= Máquinas:TamMáxBufferEnt.

{

Sistema:DiagSistema = "R8 - Os resultados da simulação mostraram melhoras no desempenho do sistema mas, algumas máquinas e os VAGs apresentam alta taxa de utilização. ";

PostMessage(Sistema:DiagSistema);

```

PostMessage( "O MA sugere que o sistema volte a ser simulado com um incremento no tamanho dos
Buffers." );
Sistema:Menu = SimÑSim3;
} );

```

```

/*****
**** RULE: ROTm9
*****/

```

```

MakeRule( ROTm9, [],
LengthList( Máquinas:ListaMáqSobreUtil ) > 0 And VAGs:UtilMédReal
> VAGs:MaxUtilVAGs And Máquinas:TamBufferEnt <= Máquinas:TamMáxBufferEnt.
{
Sistema:DiagSistema = "R9 - Os resultados da simulação não mostraram melhoras no desempenho do
sistema e algumas máquinas e os VAGs apresentam alta taxa de utilização. ";
PostMessage( Sistema:DiagSistema );
PostMessage( "O MA sugere que o sistema volte a ser simulado com um incremento no tamanho dos
Buffers." );
Sistema:Menu = SimÑSim3;
} );

```

```

/*****
**** RULE: ROTm10
*****/

```

```

MakeRule( ROTm10, [],
LengthList( Máquinas:ListaMáqSobreUtil ) > 0 And VAGs:UtilMédReal
<= VAGs:MaxUtilVAGs And Sistema:TMédCPSim < Sistema:TMédCPSimAnt
And Máquinas:TamBufferEnt <= Máquinas:TamMáxBufferEnt.
{
Sistema:DiagSistema = "R10 - Os resultados da simulação mostraram pequena melhora no desempenho
do sistema mas, algumas máquinas excederam o limite máximo de utilização. ";
PostMessage( Sistema:DiagSistema );
PostMessage( "O nº de Pallets encontra-se no seu limite máx. em relação ao nº de VAGs e ao tamanho
dos Buffers. O MA sugere que o sistema volte a ser simulado com aumento dos Buffers e redução nos
Pallets." );
Sistema:Menu = SimÑSim5;
} );

```

```

/*****
**** RULE: Mach_Local_Oper_Diag1
*****/

```

```

MakeRule( Mach_Local_Oper_Diag1, [x|Máquinas].
x:Diag_Util_Máq #!= Alta_Util And x:Diag_Fila #!= Longa.
x:Diag_Oper_Local_Máq = "IAR parece alto para esta máquina" );

```

```

/*****
**** RULE: Mach_Local_Oper_Diag2
*****/

```

```

MakeRule( Mach_Local_Oper_Diag2, [x|Máquinas].
x:Diag_Util_Máq #!= Alta_Util And ( x:Diag_Fila #!= Normal Or
x:Diag_Fila #!= Curta ).
x:Diag_Oper_Local_Máq = "IAR parece ok para esta máquina mas verifique longas operações" );

```

```

/*****
**** RULE: Mach_Local_Oper_Diag3
*****/

```



```

*****/
MakeRule( Mach_Local_Oper_Diag3, [x|Máquinas].
( x:Diag_Util_Máq #≠ Normal_Util Or x:Diag_Util_Máq #≠ Baixa_Util )
And x:Diag_Fila #≠ Longa,
x:Diag_Oper_Local_Máq = "IAR parece ok para esta máquina mas verifique a confiabilidade e tempos
de C/D");

```

```

/*****
**** RULE: Mach_Local_Oper_Diag4
*****/
MakeRule( Mach_Local_Oper_Diag4, [x|Máquinas].
x:Diag_Util_Máq #≠ Normal_Util And ( x:Diag_Fila #≠ Normal
Or x:Diag_Fila #≠
Curta ),
x:Diag_Oper_Local_Máq = "IAR parece ok para esta máquina");

```

```

/*****
**** RULE: Mach_Local_Oper_Diag5
*****/
MakeRule( Mach_Local_Oper_Diag5, [x|Máquinas].
x:Diag_Util_Máq #≠ Baixa_Util And ( x:Diag_Fila #≠ Normal
Or x:Diag_Fila #≠
Curta ),
x:Diag_Oper_Local_Máq = "IAR parece baixo para esta máquina");

```

```

/*****
**** RULE: Mach_Global_Oper_Diag1
*****/
MakeRule( Mach_Global_Oper_Diag1, [x|Máquinas].
x:Diag_Oper_Local_Máq #≠ IAR_Looks_High_for_this_Machine And
x:Diag_Util_Relativa_Máq #≠ Normal And x:Oper_Longa #≠
No.
x:Diag_Oper_Global_Máq = "IAR parece alto para esta máquina"););

```

```

/*****
**** RULE: Mach_Global_Oper_Diag2
*****/
MakeRule( Mach_Global_Oper_Diag2, [x|Máquinas].
x:Diag_Oper_Local_Máq #≠ IAR_Looks_High_for_this_Machine And
x:Diag_Util_Relativa_Máq #≠ Alta And x:Oper_Longa #≠ No.
x:Diag_Oper_Global_Máq = "Máquina desbalanceada" );

```

```

/*****
**** RULE: Mach_Global_Oper_Diag3
*****/
MakeRule( Mach_Global_Oper_Diag3, [x|Máquinas].
x:Diag_Oper_Local_Máq #≠ IAR_Looks_High_for_this_Machine And
x:Diag_Util_Relativa_Máq #≠ Alta And x:Oper_Longa #≠ Yes.
x:Diag_Oper_Global_Máq = "Máquina desbalanceada com longa operação" );

```

```

/*****
**** RULE: Mach_Global_Oper_Diag4
*****/
MakeRule( Mach_Global_Oper_Diag4, [x|Máquinas].

```

```

x:Diag_Oper_Local_Máq #= IAR_Looks_OK_for_this_Machine_but_Check_Long_Operations_Times
And x:Oper_Longa #= Yes,
x:Diag_Oper_Global_Máq = "Máquina com longa operação" );

/*****
**** RULE: Mach_Global_Oper_Diag6
*****/
MakeRule( Mach_Global_Oper_Diag6, [x|Máquinas].
x:Diag_Oper_Local_Máq #=
IAR_Looks_OK_for_this_Machine_but_Check_Reliability_and_Load_Unload_Oper_Times
And x:Diag_Confiabilidade_Máq #= Ruim And x:Diag_Tempo_CD_Máq
# = OK,
x:Diag_Oper_Global_Máq = "Máquina com baixa confiabilidade" );

/*****
**** RULE: Mach_Global_Oper_Diag5
*****/
MakeRule( Mach_Global_Oper_Diag5, [x|Máquinas].
x:Diag_Oper_Local_Máq #= IAR_Looks_OK_for_this_Machine.
x:Diag_Oper_Global_Máq = "Máquina está ok" );

/*****
**** RULE: Mach_Global_Oper_Diag7
*****/
MakeRule( Mach_Global_Oper_Diag7, [x|Máquinas].
x:Diag_Oper_Local_Máq #=
IAR_Looks_OK_for_this_Machine_but_Check_Reliability_and_Load_Unload_Oper_Times
And x:Diag_Confiabilidade_Máq #= OK And x:Diag_Tempo_CD_Máq
# = Ruim.
x:Diag_Oper_Global_Máq = "Máquina com operações de C/D, muito lentas" );

/*****
**** RULE: Mach_Global_Oper_Diag8
*****/
MakeRule( Mach_Global_Oper_Diag8, [x|Máquinas].
x:Diag_Oper_Local_Máq #= IAR_Looks_Low_for_this_Machine.
x:Diag_Oper_Global_Máq = "Máquina com baixa utilização" );

/*****
**** RULE: Mach_Action1
*****/
MakeRule( Mach_Action1, [x|Máquinas].
x:Diag_Oper_Global_Máq #= IAR parece alto para esta máquina.
x:Ação_na_Máq = "By the Simulation Results and Analysis. the Jobs IAR Looks Very High for this
Machine. Improve the Efficiency, Change the Machine. Reduce the IAR or Change the Routing." );

/*****
**** RULE: Mach_Action2
*****/
MakeRule( Mach_Action2, [x|Máquinas].
x:Diag_Oper_Global_Máq #= "Máquina desbalanceada".
x:Ação_na_Máq = "By the Simulation Results and Analysis. this Machine Looks Umbalanced. We
Suggest you to Change the Routing or Decrease the IAR." );

```

```

/*****
**** RULE: Mach_Action3
*****/
MakeRule( Mach_Action3, [x|Máquinas],
  x:Diag_Oper_Global_Máq #="Máquina desbalanciada com longa operação",
  x:Ação_na_Máq = "By the Simulation Results and Analysis, this Machine Looks Unbalanced, but it
make a Long Operation. We Suggest you to Change the Routing or Change the Machine Efficiency." );

/*****
**** RULE: Mach_Action4
*****/
MakeRule( Mach_Action4, [x|Máquinas],
  x:Diag_Oper_Global_Máq #="Máquina com longa operação",
  x:Ação_na_Máq = "By the simulation results and analysis, this machine has a high utilization due to
long time operations. If this are necessary we suggest you to change the machine efficiency." );

/*****
**** RULE: Mach_Action5
*****/
MakeRule( Mach_Action5, [x|Máquinas],
  x:Diag_Oper_Global_Máq #="Máquina está ok".
  x:Ação_na_Máq = "By the simulation results and analysis, this machine looks OK. There is no action
at this moment." );

/*****
**** RULE: Mach_Action6
*****/
MakeRule( Mach_Action6, [x|Máquinas],
  x:Diag_Oper_Global_Máq #="Máquina com baixa confiabilidade".
  x:Ação_na_Máq = "By the simulation results and analysis, this machine looks unreliable. We suggest
you to change this machine or improve the maintenance system." );

/*****
**** RULE: Mach_Action7
*****/
MakeRule( Mach_Action7, [x|Máquinas],
  x:Diag_Oper_Global_Máq #="Máquina com operações de C/D, muito lentas",
  x:Ação_na_Máq = "By the simulation results and analysis, the AGVs/Machines Load/Unload times
looks very slow. We suggest you to improve the AGVs/Machines Load/Unload system." );

/*****
**** RULE: Mach_Action8
*****/
MakeRule( Mach_Action8, [x|Máquinas],
  x:Diag_Oper_Global_Máq #="Máquina com baixa utilização".
  x:Ação_na_Máq = "By the simulation results and analysis, this machine has been low utilized. We
suggest you to increase the IAR or change the routing." );

/*****
**** RULE: AGVs_Diag1
*****/
MakeRule( AGVs_Diag1, [],
  AGVs:Util_Diag #="Alta_Util And AGVs:Diag_Fila #=" Longa.
  AGVs:AGV_Diag = Num_of_AGVs_Looks_Low );

```

```

/*****
*** RULE: AGVs_Diag2
*****/
MakeRule( AGVs_Diag2, [],
AGVs:Util_Diag #!= Alta_Util And ( AGVs:Diag_Fila #!= Normal
Or AGVs:Diag_Fila #!=
Curta ),
AGVs:AGV_Diag = Num_of_AGVs_Looks_Low_But_Check_Speed_and_Pallets );

/*****
*** RULE: AGVs_Diag3
*****/
MakeRule( AGVs_Diag3, [],
AGVs:Util_Diag #!= Normal_Util And AGVs:Diag_Fila #!= Longa.
AGVs:AGV_Diag = Num_of_AGVs_Looks_OK_but_Check_BreakDowns );

/*****
*** RULE: AGVs_Diag5
*****/
MakeRule( AGVs_Diag5, [],
AGVs:Util_Diag #!= Baixa_Util And AGVs:Diag_Fila #!= Longa.
AGVs:AGV_Diag = Num_of_AGVs_Looks_High_but_Check_BreakDowns );

/*****
*** RULE: AGVs_Diag4
*****/
MakeRule( AGVs_Diag4, [],
AGVs:Util_Diag #!= Normal_Util And ( AGVs:Diag_Fila #!= Normal
Or AGVs:Diag_Fila
#!= Curta ),
AGVs:AGV_Diag = Num_of_AGVs_Looks_OK );

/*****
*** RULE: AGVs_Diag6
*****/
MakeRule( AGVs_Diag6, [],
AGVs:Util_Diag #!= Baixa_Util And ( AGVs:Diag_Fila #!= Normal
Or AGVs:Diag_Fila #!=
Curta ).
AGVs:AGV_Diag = High_Num_of_AGVs );

/*****
*** RULE: AGVs_Global_Diag1
*****/
MakeRule( AGVs_Global_Diag1, [],
AGVs:AGV_Diag #!= Num_of_AGVs_Looks_Low And AGVs:Speed_Diag
#!= OK And Estrados:Pallets_Diag #!= Num_of_Pallets_Looks_OK.
AGVs:AGV_Global_Diag = Low_Num_of_AGVs );

/*****
*** RULE: AGVs_Global_Diag2
*****/
MakeRule( AGVs_Global_Diag2, [],

```

```
AGVs:AGV_Diag #≠ Num_of_AGVs_Looks_Low And AGVs:Speed_Diag
#≠ Ruim And Estrados:Pallets_Diag #≠ Num_of_Pallets_Looks_OK,
AGVs:AGV_Global_Diag = AGVs_Speed_Looks_Low );
```

```
/******
**** RULE: AGVs_Global_Diag3
*****/
```

```
MakeRule( AGVs_Global_Diag3, [],
AGVs:AGV_Diag #≠ Num_of_AGVs_Looks_Low_But_Check_Speed_and_Pallets
And AGVs:Speed_Diag #≠ Ruim And Estrados:Pallets_Diag
#≠ Num_of_Pallets_Looks_OK,
AGVs:AGV_Global_Diag = AGVs_Speed_Looks_Low );
```

```
/******
**** RULE: AGVs_Global_Diag4
*****/
```

```
MakeRule( AGVs_Global_Diag4, [],
AGVs:AGV_Diag #≠ Num_of_AGVs_Looks_Low_But_Check_Speed_and_Pallets
And AGVs:Speed_Diag #≠ OK And Estrados:Pallets_Diag #≠
Num_of_Pallets_Looks_Low,
AGVs:AGV_Global_Diag = Low_Num_of_AGVs );
```

```
/******
**** RULE: AGVs_Global_Diag5
*****/
```

```
MakeRule( AGVs_Global_Diag5, [],
AGVs:AGV_Diag #≠ Num_of_AGVs_Looks_OK_but_Check_BreakDowns
And AGVs:AGV_Reliability_Diag #≠ Ruim.
AGVs:AGV_Global_Diag = AGVs_Unreliables );
```

```
/******
**** RULE: AGVs_Global_Diag6
*****/
```

```
MakeRule( AGVs_Global_Diag6, [],
AGVs:AGV_Diag #≠ Num_of_AGVs_Looks_OK_but_Check_BreakDowns
And AGVs:AGV_Reliability_Diag #≠ OK.
AGVs:AGV_Global_Diag = Possible_Slow_Load_Unload_Process );
```

```
/******
**** RULE: AGVs_Global_Diag7
*****/
```

```
MakeRule( AGVs_Global_Diag7, [],
AGVs:AGV_Diag #≠ Num_of_AGVs_Looks_OK.
AGVs:AGV_Global_Diag = AGVs_are_OK );
```

```
/******
**** RULE: AGVs_Global_Diag8
*****/
```

```
MakeRule( AGVs_Global_Diag8, [],
AGVs:AGV_Diag #≠ Num_of_AGVs_Looks_High_but_Check_BreakDowns
And AGVs:AGV_Reliability_Diag #≠ OK.
AGVs:AGV_Global_Diag = Possible_Slow_Load_Unload_Process );
```

```
/******
```

```

**** RULE: AGVs_Global_Diag9
*****/
MakeRule( AGVs_Global_Diag9, [],
AGVs:AGV_Diag #= Num_of_AGVs_Looks_High_but_Check_BreakDowns
And AGVs:AGV_Reliability_Diag #= Ruim,
AGVs:AGV_Global_Diag = AGVs_Unreliables );

/****
**** RULE: AGVs_Global_Diag10
*****/
MakeRule( AGVs_Global_Diag10, [],
AGVs:AGV_Diag #= High_Num_of_AGVs,
AGVs:AGV_Global_Diag = Num_AGVs_are_High );

/****
**** RULE: AGVs_Action1
*****/
MakeRule( AGVs_Action1, [],
AGVs:AGV_Global_Diag #= Low_Num_of_AGVs.
AGVs:AGV_Action = "By the Simulation Results and Analysis. the Number of AGVs Looks Low . We
Suggest you to Increase the Number of AGVs" );

/****
**** RULE: AGVs_Action2
*****/
MakeRule( AGVs_Action2, [],
AGVs:AGV_Global_Diag #= AGVs_Speed_Looks_Low.
AGVs:AGV_Action = "By the Simulation Results and Analysis. the Speed of the AGVs Looks Low.
We Suggest you to Increase the Speed of the AGVs" );

/****
**** RULE: AGVs_Action3
*****/
MakeRule( AGVs_Action3, [],
AGVs:AGV_Global_Diag #= AGVs_Unreliables.
AGVs:AGV_Action = "By the Simulation Results and Analysis. the AGVs Looks Unreliables. We
Suggest you to change the AGVs or Improve your Maintenance System" );

/****
**** RULE: AGVs_Action4
*****/
MakeRule( AGVs_Action4, [],
AGVs:AGV_Global_Diag #= Possible_Slow_Load_Unload_Process.
AGVs:AGV_Action = "By the Simulation Results and Analysis. your AGVs/Machines Load Unload
Process is Looks Slow. We Suggest you to Improve the Load Unload Process" );

/****
**** RULE: AGVs_Action5
*****/
MakeRule( AGVs_Action5, [],
AGVs:AGV_Global_Diag #= AGVs_are_OK.
AGVs:AGV_Action = "By the Simulation Results and Analysis. the AGVs System Looks OK. There is
No Action at this Moment." );

```

```

/*****
**** RULE: AGVs_Action6
*****/
MakeRule( AGVs_Action6, [],
  AGVs:AGV_Global_Diag #= Num_AGVs_are_High,
  AGVs:AGV_Action = "By the Simulation Results and Analysis, the Number of AGVs Looks High. We
  Suggest you to Reduce the Number of AGVs" );

/*****
**** RULE: Pallets_Diag1
*****/
MakeRule( Pallets_Diag1, [],
  Estrados:Pallets_Util_Diag #= Alta_Util And Estrados:Diag_Fila
  #= Longa,
  Estrados:Pallets_Diag = Num_of_Pallets_Looks_Low );

/*****
**** RULE: Pallets_Diag2
*****/
MakeRule( Pallets_Diag2, [],
  Estrados:Pallets_Util_Diag #= Normal_Util And Estrados:Diag_Fila
  #= Longa,
  Estrados:Pallets_Diag = Num_of_Pallets_Looks_OK_But_Check_LU_Operations );

/*****
**** RULE: Pallets_Diag5
*****/
MakeRule( Pallets_Diag5, [],
  Estrados:Pallets_Util_Diag #= Baixa_Util And ( Estrados:Diag_Fila
  #= Normal
  Or Estrados:Diag_Fila
  #= Curta ).
  Estrados:Pallets_Diag = Num_of_Pallets_Looks_High );

/*****
**** RULE: Pallets_Diag3
*****/
MakeRule( Pallets_Diag3, [],
  Estrados:Pallets_Util_Diag #= Normal_Util And ( Estrados:Diag_Fila
  #= Normal
  Or Estrados:Diag_Fila
  #= Curta ).
  Estrados:Pallets_Diag = Num_of_Pallets_Looks_OK );

/*****
**** RULE: Pallets_Diag4
*****/
MakeRule( Pallets_Diag4, [],
  Estrados:Pallets_Util_Diag #= Baixa_Util And Estrados:Diag_Fila
  #= Longa,
  Estrados:Pallets_Diag = Num_of_Pallets_Looks_High_But_Check_LU_Operations );

/*****
**** RULE: Pallets_Action1

```

```

*****/
MakeRule( Pallets_Action1, [],
  Estrados:Pallets_Diag #= Num_of_Pallets_Looks_Low.
  Estrados:Pallets_Action = "By the simulation results and analysis, the number of Pallets looks low. We
suggest you to increase the number of Pallets." );

/*****
**** RULE: Pallets_Action2
*****/
MakeRule( Pallets_Action2, [],
  Estrados:Pallets_Diag #= Num_of_Pallets_Looks_OK_But_Check_LU_Operations
  And Estrados:Pallets_LU_Diag #= Ruim,
  Estrados:Pallets_Action = "By the simulation results and analysis, the number of Pallets looks OK. but
the Pallets Load/Unload operations looks slow. We suggest you to improve the these operations." );

/*****
**** RULE: Pallets_Action5
*****/
MakeRule( Pallets_Action5, [],
  Estrados:Pallets_Diag #= Num_of_Pallets_Looks_High.
  Estrados:Pallets_Action = "By the simulation results and analysis, the number of Pallets looks high. We
suggest you to reduce the number of Pallets." );

/*****
**** RULE: Pallets_Action3
*****/
MakeRule( Pallets_Action3, [],
  Estrados:Pallets_Diag #= Num_of_Pallets_Looks_OK.
  Estrados:Pallets_Action = "By the simulation results and analysis, the number of Pallets looks OK.
There is no actions at this moment." );

/*****
**** RULE: Pallets_Action4
*****/
MakeRule( Pallets_Action4, [],
  Estrados:Pallets_Diag #= Num_of_Pallets_Looks_High_But_Check_LU_Operations
  And Estrados:Pallets_LU_Diag #= Ruim,
  Estrados:Pallets_Action = "By the simulation results and analysis, the number of Pallets looks high but
the Pallets Load/Unload operations looks slow. We suggest you to first improve the these operations." );

```


ANEXO 3

PRINCIPAIS TELAS DO PROTÓTIPO

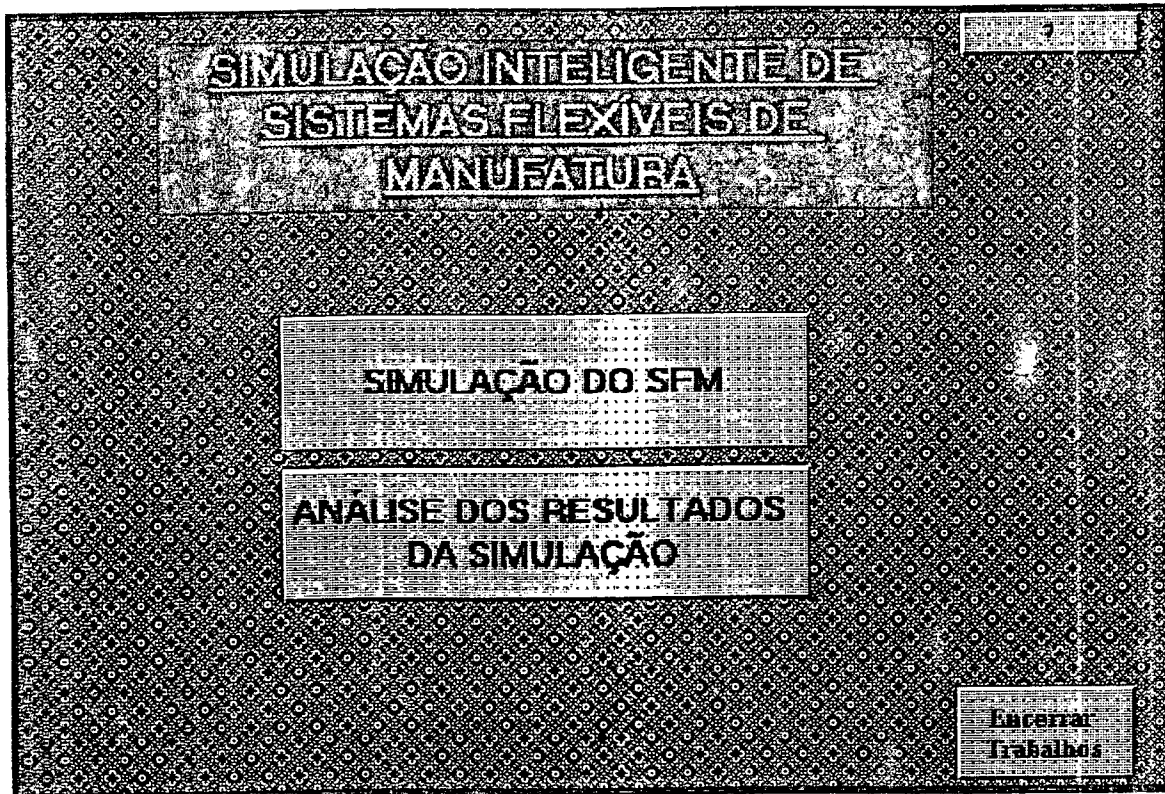


Figura A.1: Tela inicial do protótipo com alguns dos elementos da interface gráfica (GUI)

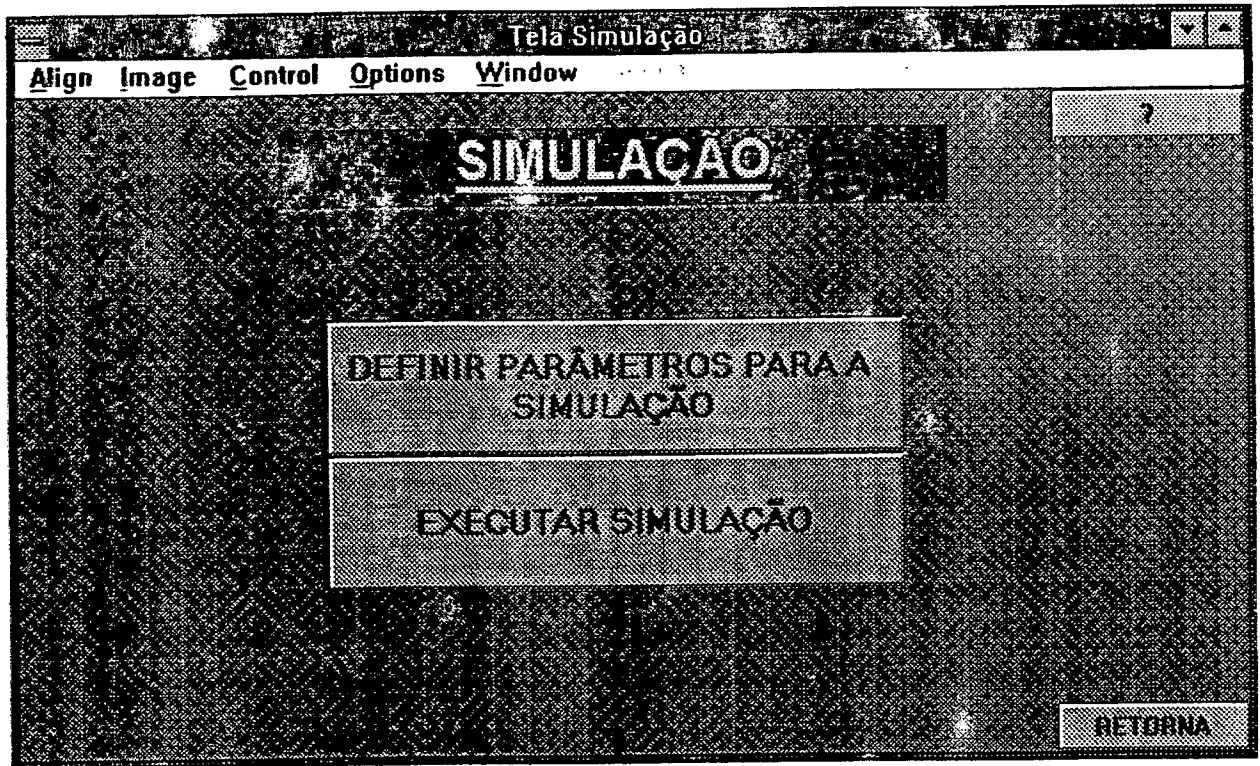


Figura A.2: Tela inicial do Módulo de Simulação com suas duas opções.

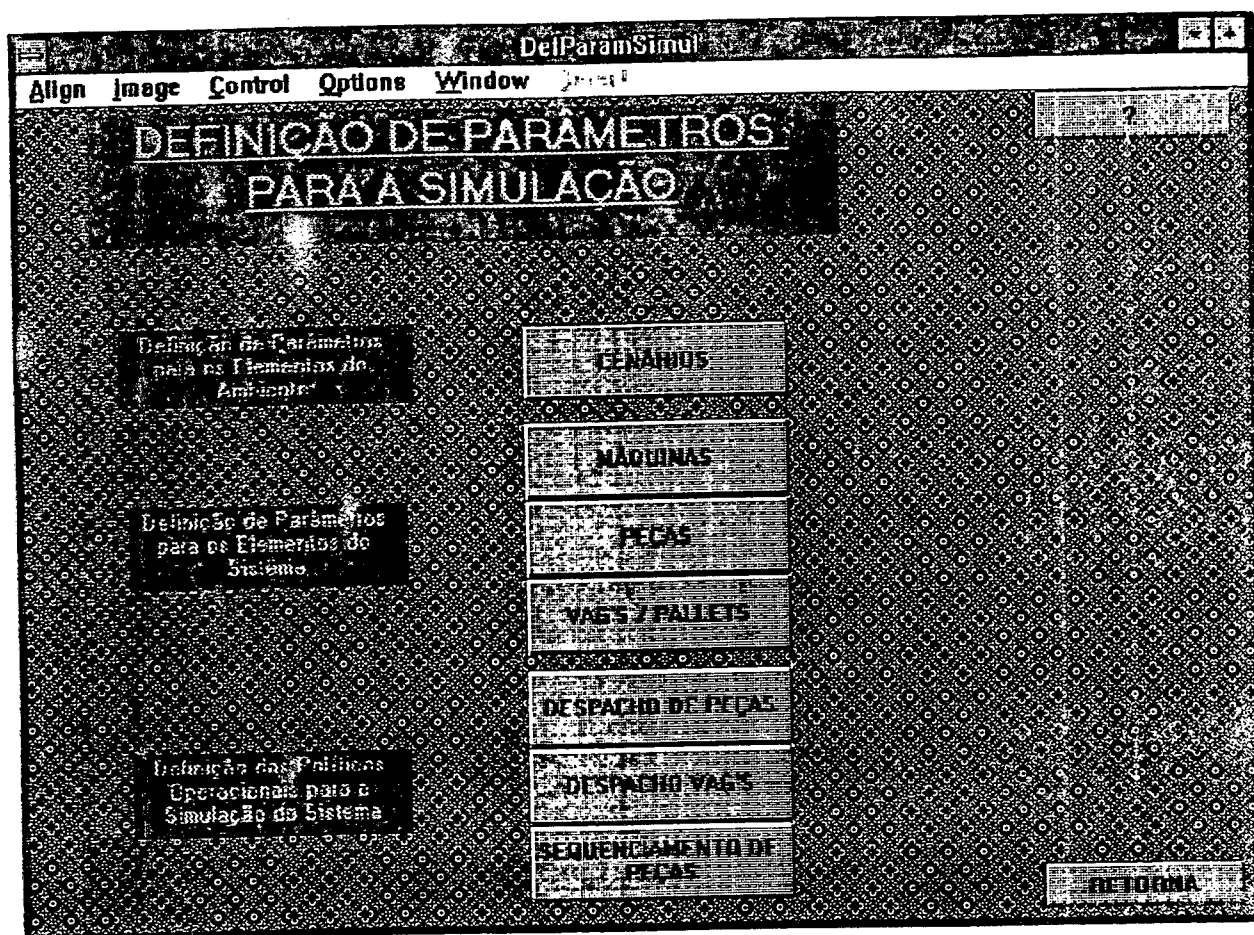


Figura A.3: Tela que permite ao usuário a definição dos parâmetros e políticas operacionais a serem empregados durante a simulação do sistema.

DefParamSimul

Align Image Control Options Window

DEFINIÇÃO DE PARÂMETROS PARA A SIMULAÇÃO

Entre os Parâmetros para a Simulação do Cenário

Número de Peças: 5

Tipos de Peças: 13, 17

Quantidade Total de Peças: 108

Quantidade de Peças por Tipo: 24, 18

3, 12, 13, 14, 15, 16, 17

OK Reset

SEQUENCIAMENTO DE PEÇAS

RETORNA

Figura A.4: Exemplo da definição de um cenário. Nesta tela são definidas quais peças serão fabricadas e em qual volume.

The screenshot shows a software window titled "DefParamSimul" with a menu bar containing "Align", "Image", "Control", "Options", "Window", and "Help". The main area has a title "DEFINIÇÃO DE PARÂMETROS PARA A SIMULAÇÃO" and a subtitle "Digite os Parâmetros para a Máquina". A list of operations is shown with checkboxes and a scrollable list. Below the list are input fields for "Eficiências da Máquina", "Mínima Utiliz. Desejada", "Máxima Utiliz. Permitida", and "Tamanho do Buffer (Inicial)". At the bottom are buttons for "OK", "Reset", "PEÇAS", and "RETORNA".

Operação	Definir	Definir	Definir	Definir
2	<input type="checkbox"/>			
4	<input type="checkbox"/>			
Eficiências da Máquina				
0	<input type="checkbox"/>		0 .. 1	
1	<input type="checkbox"/>			
Mínima Utiliz. Desejada				
.75			0.0 .. 1.0	
Máxima Utiliz. Permitida				
.90			0.0 .. 1.0	
Tamanho do Buffer (Inicial)				
2				

OK Reset

PEÇAS RETORNA

Figura A.5: A tela acima permite a definição dos parâmetros de cada uma das máquinas. Podemos definir quais operações que cada uma realizará, as eficiências destas operações, as taxas de utilização máxima e mínima e o tamanho dos buffers.

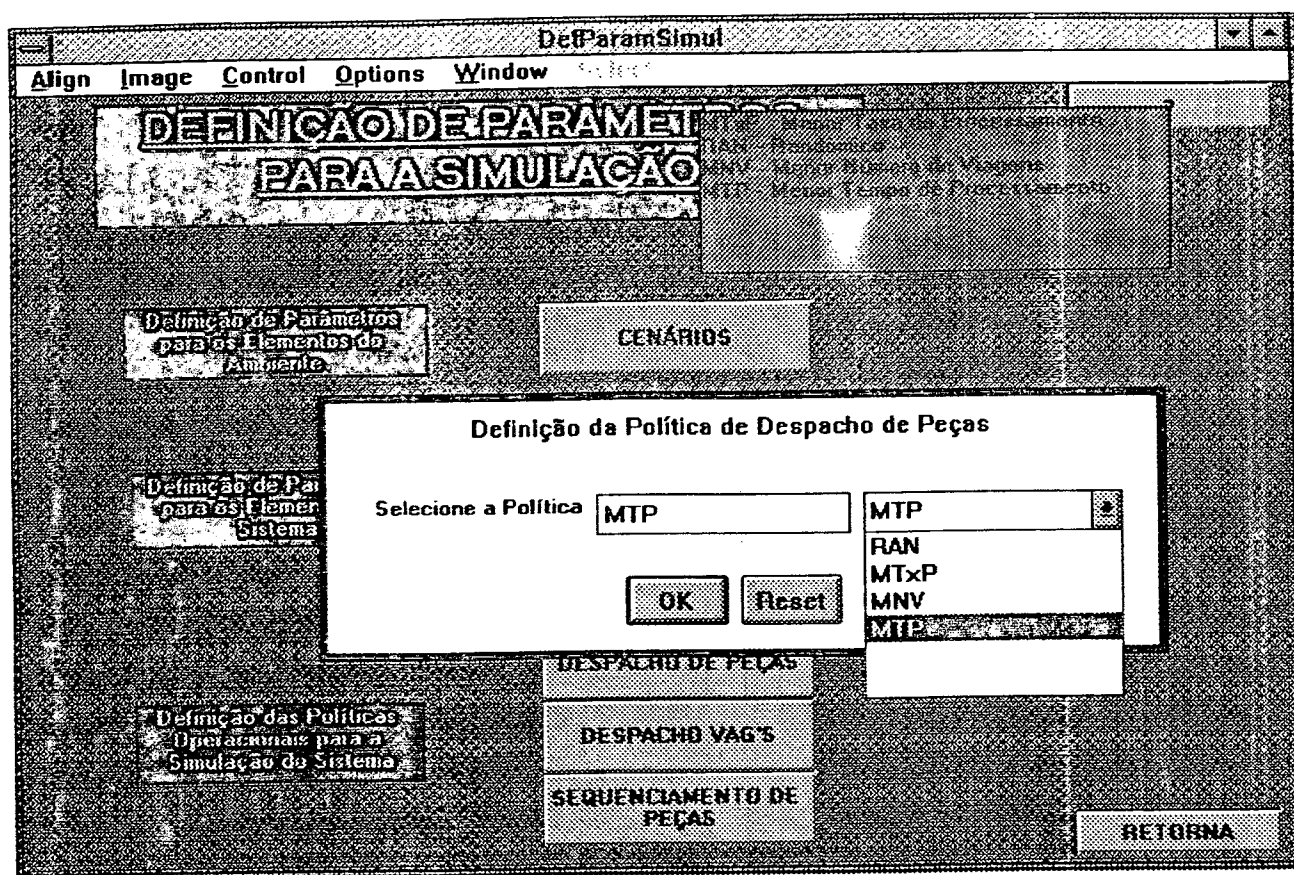


Figura A.6: Na tela acima apresentamos um exemplo de definição de políticas operacionais. No caso, a política de despacho de peças

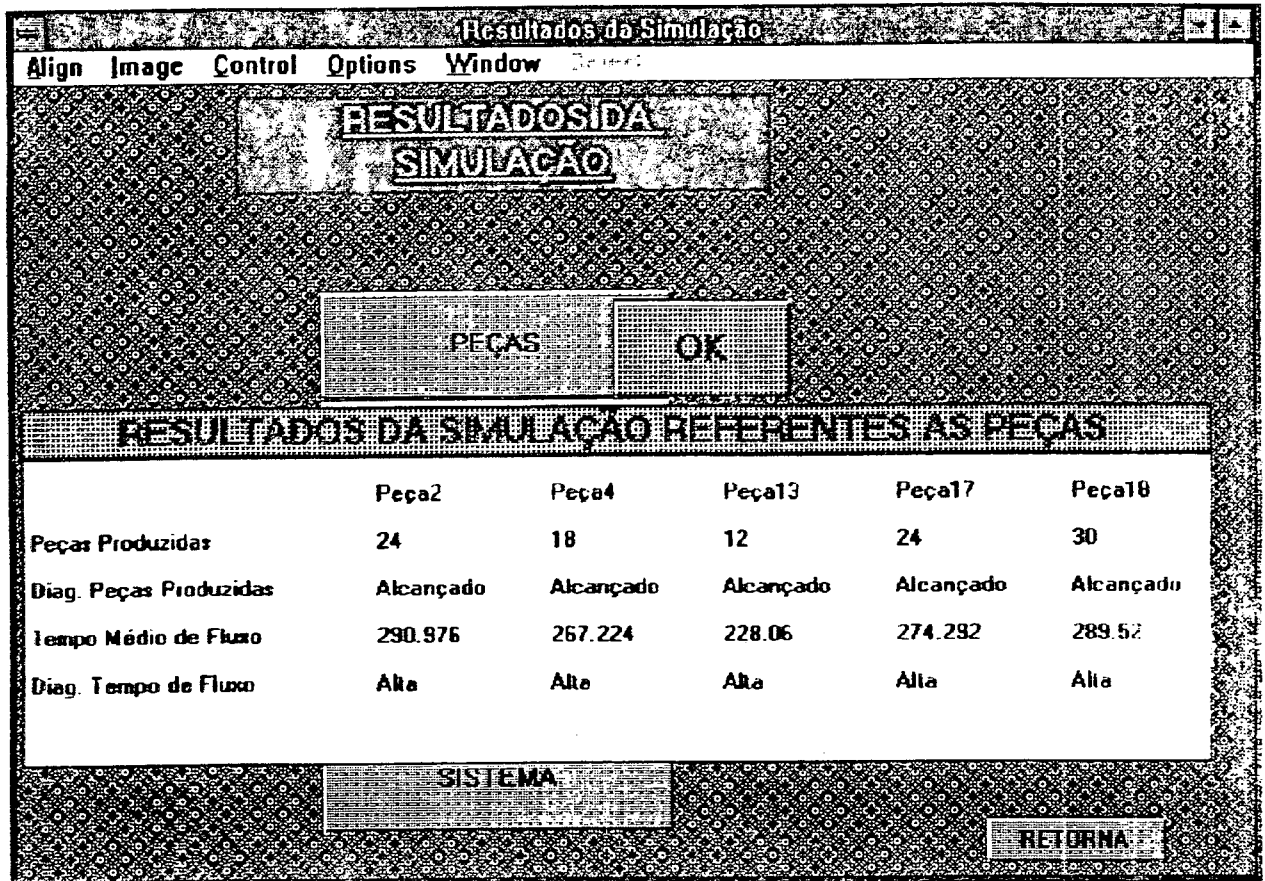


Figura A.7: A tela acima mostra um exemplo de apresentação dos resultados da simulação . No caso os resultados reference as peças produzidas.

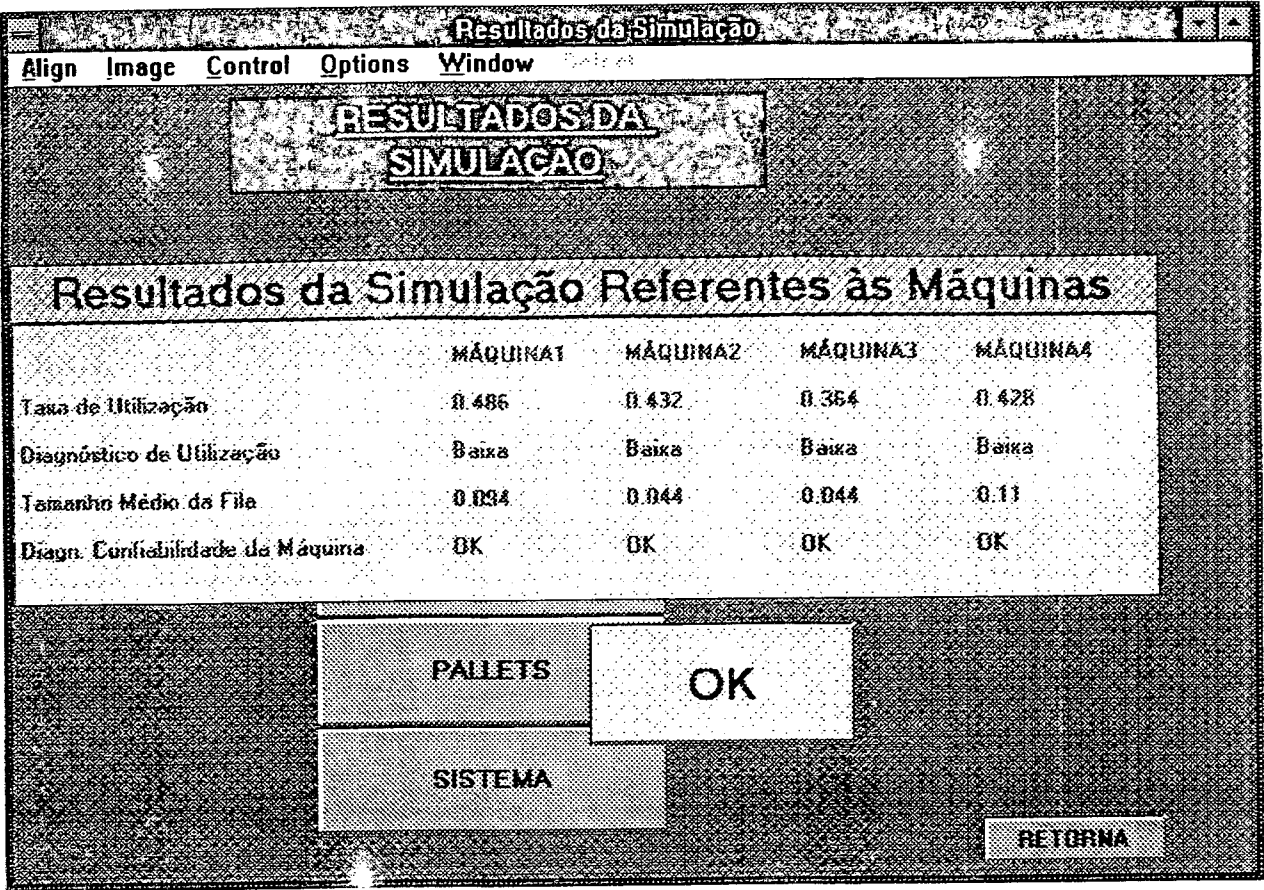


Figura A.8: Tela de apresentação dos resultados da simulação referente ao desempenho das máquinas.

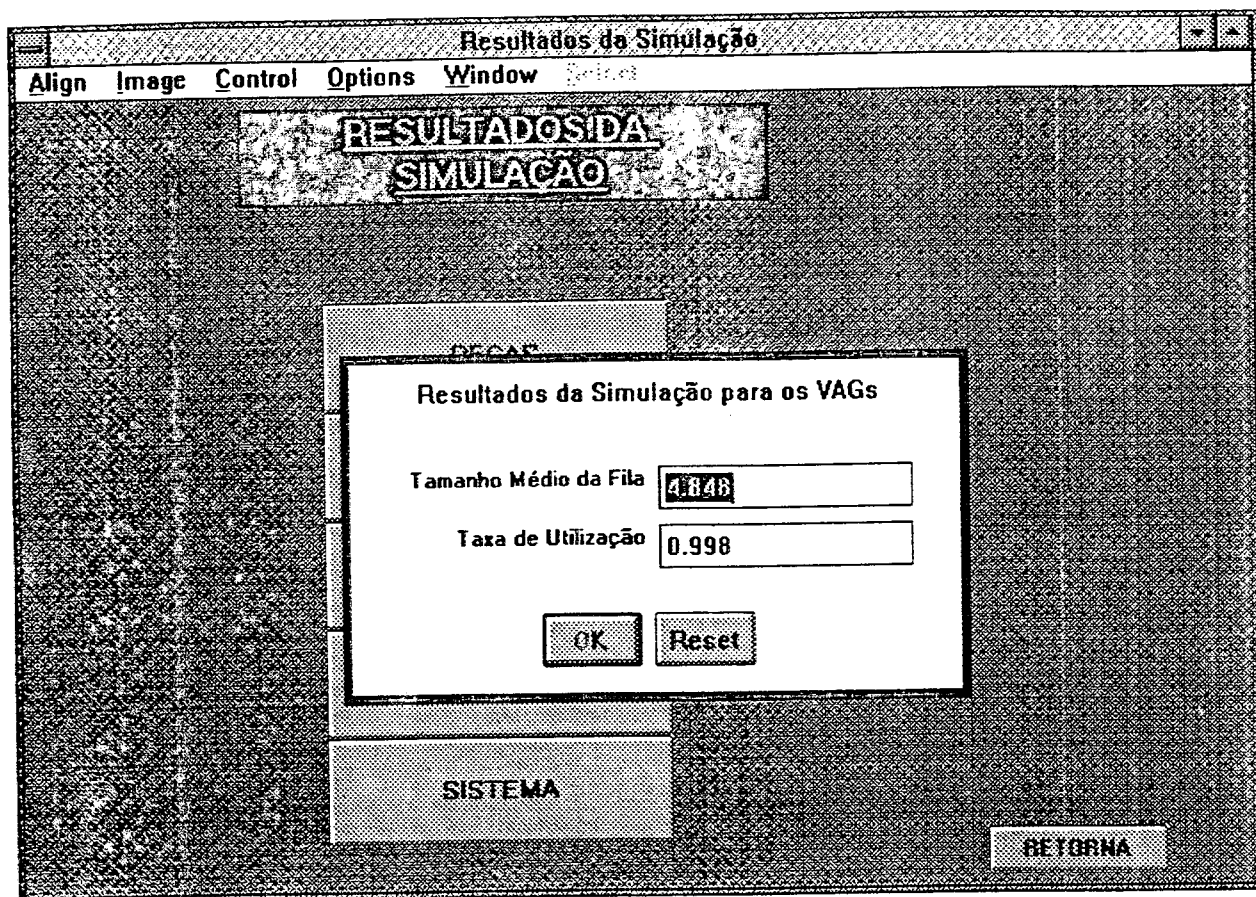


Figura A.9: Tela de apresentação dos resultados da simulação referente ao desempenho dos VAGs

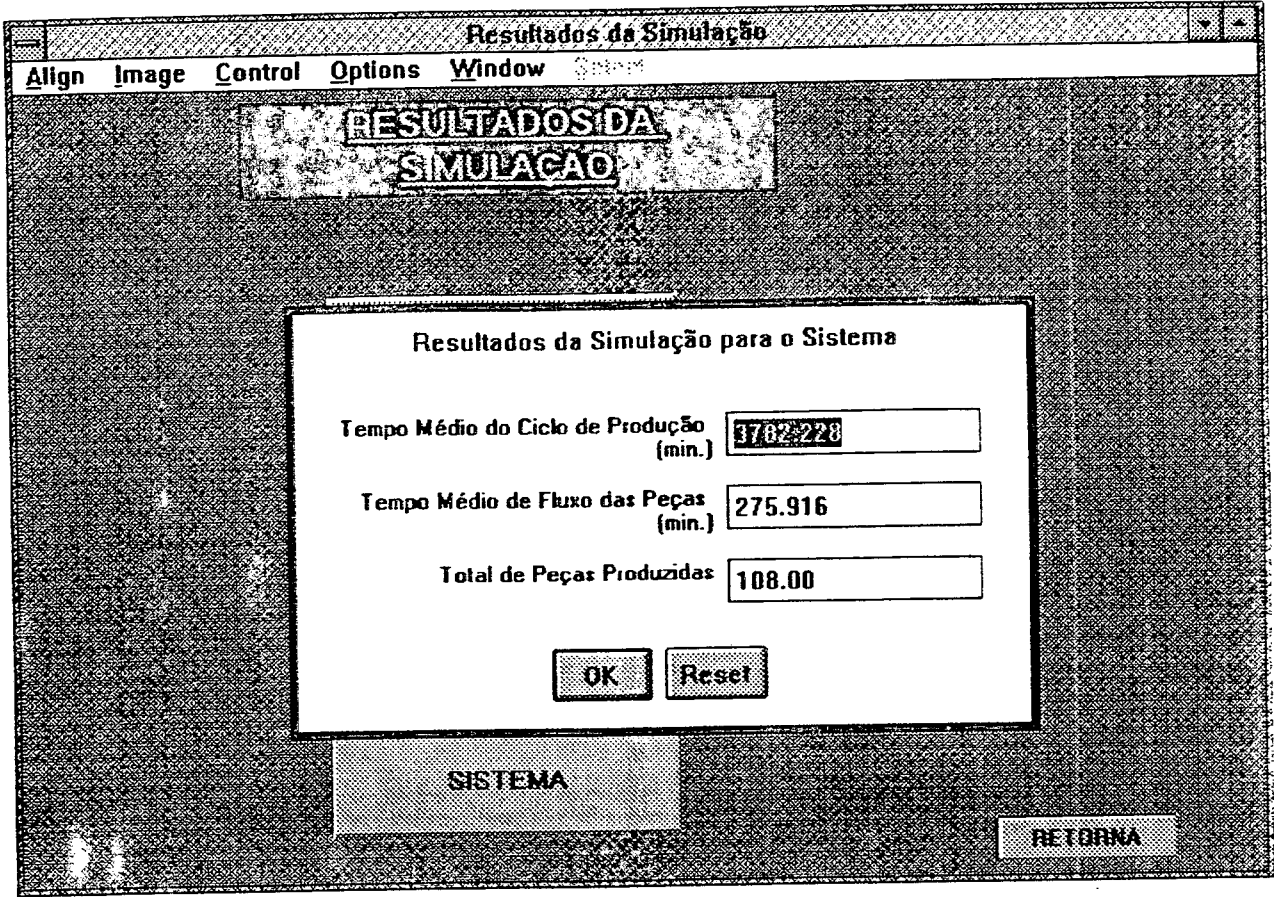


Figura A.10: Tela de apresentação dos resultados da simulação mostrando duas das principais medidas de desempenho: Tempo Médio do Ciclo de Produção e Tempo Médio do Fluxo de Peças



Figura A.11: Tela inicial do Módulo de Análise. A última opção permite iniciar o processo de análise utilizando o MA.

DefParamAnálise

Align Image Control Options Window

SESSÃO DE ANÁLISE

Definição de Parâmetros para Iniciar o Processo Iterativo de Simulação/Análise

Tamanho máximo do Buffer

Máxima Taxa de Utilização dos VAGs: 0.0 .. 1.0

Nº Mínimo de Pallets

Nº Máximo de Pallets

Máximo Tempo Estimado para o Ciclo Produtivo

OK Reset

Inicia Análise

Retorna

Figura A.12: A tela acima permite ao usuário definir alguns parâmetros básicos para iniciar o processo de análise.

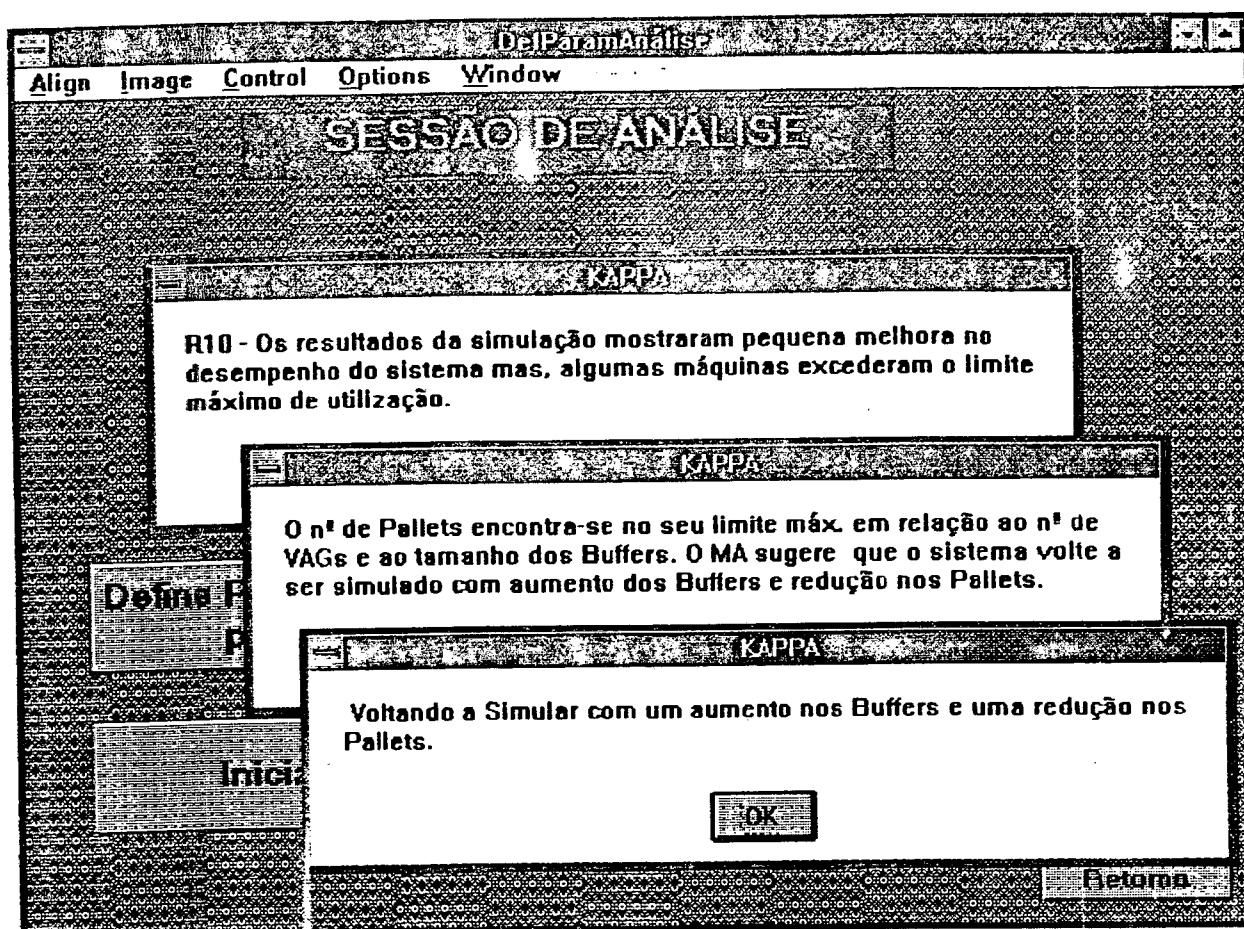


Figura A.13: A tela acima é um exemplo dos resultados das análises efetuadas pelo MA. Neste exemplo o MA sugere a continuidade das simulações com alterações nos parâmetros do modelo de simulação.

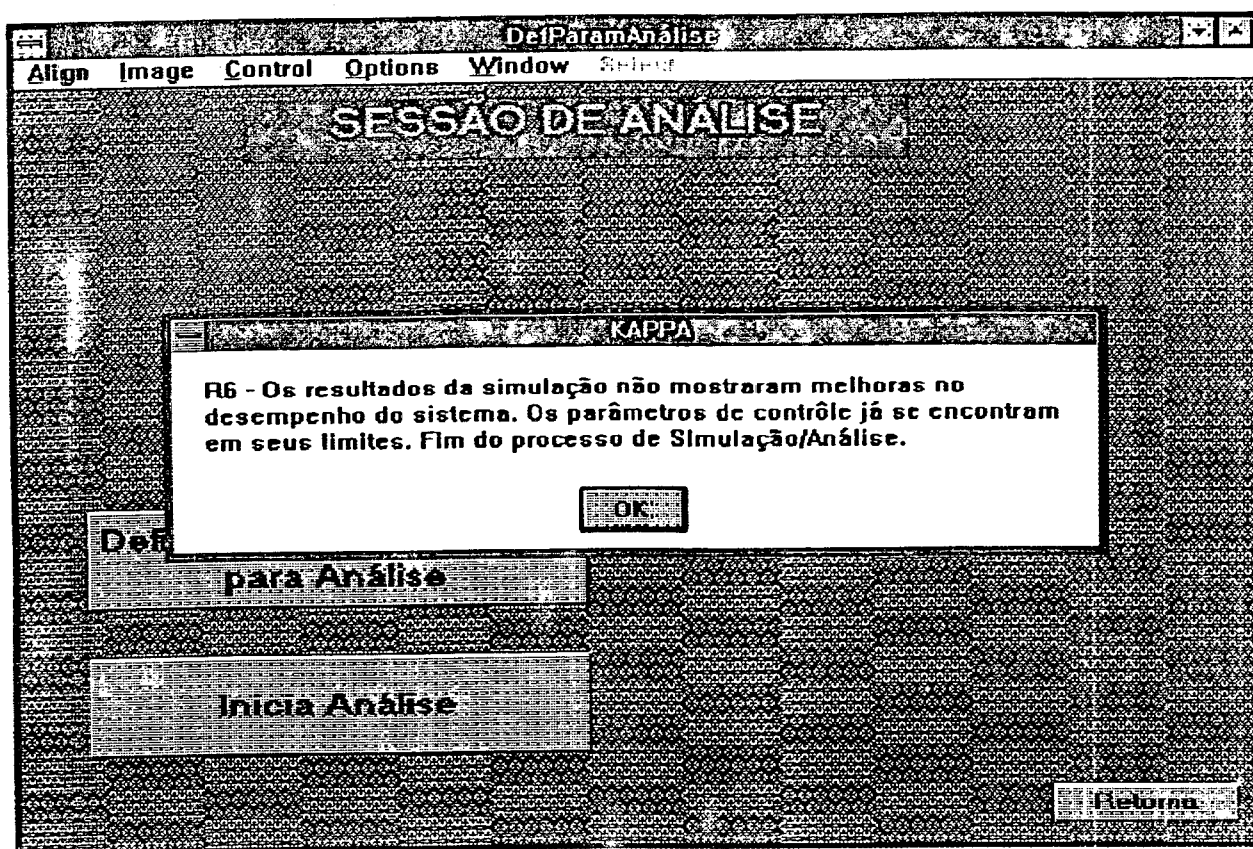


Figura A.14: A tela acima é outro exemplo dos resultados das análises efetuadas pelo MA. Nesta, o módulo de análise está sugerindo o encerramento do processo iterativo.

ANEXO 4

CÓDIGO FONTE DO MODELO DE SIMULAÇÃO
EM SIMAN IV

BEGIN;

```

CREATE;          *** cria uma entidade para def. do
ASSIGN: MR(pallet)= x(3):! N§ pallets lido em cprime
      NMaxEnt = x(7):! N§ Max de entidades (pecas)
      NPecas=x(5)+1:!   N§ (tipos) de Pecas      ****
      IT(1,1) = x(8):! IT e'uma variavel do SIMAN que identifica
      IT(1,2) = x(9):! o status de uma unidade de um transporte.
      IT(1,3) = x(10):! P.ex. IT(1,3) indica o estado 0,1 ou 2,
      IT(1,4) = x(11):! i.e'. vazio, ocupado ou inativo, da 3ª
      Dispose; unidade do transporte 1 (AGV).

```

```

CREATE,54; *** 36;          *** cria o lote de pecas
      colocando-o a disposicao
      na estacao L/U ***
CREATE,54,800; *** 36,1200;      *** segundo lote de producao ***

;
;
;
CREATE,36,2400;      *** terceiro lote *****
;
;
CREATE,27,1350;      *** quarto lote *****
;

```

```

ASSIGN: PT=UF(0); !*** User function define tipo de peca

```

```

; **** LOAD STATION ****
;
;

```

```

      STATION,      STALOAD;
      ASSIGN:      OPNUM = 1; *** seta 1ª operacao ***
      QUEUE,      LOADQUEUE;
      SEIZE:      PALLET;
      DELAY:      1: MARK(TIMEIN); *** Load PALLET
NEXTOP ASSIGN:      d(1)=PT:! *** variavel global d(1)= tipo peca
      d(2)=OPNUM; *** variavel global d(2)= tipo operacao
      EVENT:      1; *** determina operacao e melhor maquina
      p/ faze-la ***
;
      BRANCH,1:
      IF, d(3)= M, MANTEM;
      IF, d(3)= 1, GOTOM1;! *** variavel global d(3)= destino da
      IF, d(3)= 2, GOTOM2;! peca no sistema (maquinas ou saida)
      IF, d(3)= 3, GOTOM3;! o valor e'setado em EVENT 1 ***
      IF, d(3)= 4, GOTOM4;
      IF, d(3)= 6, GOTOEX;

```



```

MANTENM INSERT: MACHQ,1:
GOTOM1 ASSIGN: DEST = 1: NEXT(AGVQ):
GOTOM2 ASSIGN: DEST = 2: NEXT(AGVQ):
GOTOM3 ASSIGN: DEST = 3: NEXT(AGVQ):
GOTOM4 ASSIGN: DEST = 4: NEXT(AGVQ):
GOTOEX ASSIGN: DEST = 6: NEXT(AGVQ):
:
AGVQ QUEUE. AGVQUEUEU ::Mark(TFAGV):
REQUEST: AGV (SDS):
:Tally: TempoFilaAGV. Int(TFAGV):
DELAY: 1 ::mark(ttransp): *** Load AGV
TRANSPORT: AGV, DEST:
:
*** STATIONS 1 TO 4 ****
:
STATION. STA1-STA4:
DELAY: 1; *** Unload AGV
:TALLY: PT+NPECas, int(ttransp): *** Tallies Pt(1,2...)+NPECas(N§ de pecas) **
:tally: TTranspTotal, int(ttransp):
BRANCH,1:
IF, NQ(M).GE.x(1), contagv:
ELSE, Lqscan;
contagv COUNT: 1; *** conta n§ vezes AGV fica bloqueado
em frente das maquinas ***
:
Lqscan QUEUE. (m+8);
SCAN: NQ(m)< x(1): *** x(1)= Espaco disponivel no buffer
entrada. x(1) e'lido de arquivo
de dados gerado pelo Kappa. O bloco
SCAN segura o AGV em frente a maq.
ate' que exista espaco no buffer **
:
BRANCH,1:
IF, NQ(AGVQUEUE) > 0. FREEAGV:
ELSE, GOSTAGE;
:
FREEAGV FREE: AGV;
:
MACHQ QUEUE. M ::Mark(TFila):
SEIZE: MACH(M):
:Tally: PT + 2*NPECas, int(TFila):
:Tally: TEspTotal . int(TFila):
ASSIGN: d(5)=M:
d(2)=OpNum:
d(1)=PT;
EVENT: 2; *** determina tempo de operacao a
partir de d(1), d(2) e d(5) ***
DELAY: d(4); *** d(4) recebe o tempo da operacao
em EVENT 2 ***
:
BRANCH,1:
IF.NQ(M+4) == x(2),LALT:!! *** x(2)=Tamanho Buffer saida
ELSE.LREL; este BRANCH confirma ou nao a
indisponibilidade (bloqueio) da

```



```

      maquina, enquanto o buffer de
      saida estiver lotado. ***
:
:
:
LALT  ALTER:    MACH(M),-1;
      COUNT:    2;      *** conta maquinas bloqueadas pelo
                        buffer de saida ***
:
      QUEUE,    altQ;
      SCAN:     NQ(M+4) < x(2);
      ALTER:    MACH(M),+1;
LREL  RELEASE:  MACH(M);
      ASSIGN:    OpNum = OpNum+1: NEXT (NEXTOP);
GOSTAGE BRANCH,2:
      ALWAYS, GOBACK;
      ALWAYS, MACHQ;
:
GOBACK MOVE:    AGV, STASTAGE : NEXT (FREEAGV2);
:
:
*** UNLOAD STATION ***
:
      STATION, STAUNLOAD;
      RELEASE:  PALLET;
      DELAY:    1;    *** Unload PALLET
TALLY:  PT, INT(TIMEIN);    ** tallies de 1 ate n§ de pecas
TALLY:  TFluxoTot, INT(TIMEIN);
      ASSIGN:  NEnt = NEnt +1;
      BRANCH,1:
          IF, NQ(AGVQUEUE) > 0, FREEAGV2;
          ELSE, GOBACK;
:
FREEAGV2 FREE:    AGV:DISPOSE:
:
*****
* Ultima Entidade Printa os Resultados em Arquivo para Leitura pelo Kappa *
*****
      CREATE:
      QUEUE,    PRINTQ;
      SCAN:     NEnt.EQ.NMaxEnt;
:
      ASSIGN:  NTal1 = NPecas;
GOTOUP1 ASSIGN: ind1 = ind1+1;
      BRANCH,1:
          IF, ind1 <= NTal1, PRINT1;
          ELSE, GoDow1;
PRINT1  WRITE, OUT, "(2F7.2)": TAVG(ind1), TNUM(ind1) : NEXT(GOTOUP1);
:
:
GoDow1  ASSIGN: NTal2 = (2*NPecas);
GOTOUP2 BRANCH,1:
:
      IF, ind1 <= NTal2, PRINT2;
      ELSE, GoDow2;
PRINT2  WRITE, OUT, "(F7.2)": TAVG(ind1);

```

```

:      ASSIGN: ind1 = ind1+1:NEXT(GOTOUP2);
:
:
:
:GoDow2  ASSIGN: NTal3=(2*NPecas)+2;
:GOTOUP3  BRANCH,1:
:      IF, ind1 <= NTal3, PRINT3:
:      ELSE, GoDow3:
:PRINT3  WRITE, OUT, "(2F7.2)": TAVG(ind1), TNUM(ind1);
:      ASSIGN: ind1 = ind1+1:NEXT(GOTOUP3);
:
:
:
GoDow1  ASSIGN: NTal2 = 1+NPecas;
GOTOUP2  BRANCH,1:
:      IF, ind1 <= NTal2, PRINT2:
:      ELSE, Dstats;
PRINT2  WRITE, OUT, "(2F7.2)": TAVG(ind1), TNUM(ind1);
:      ASSIGN: ind1 = ind1+1:NEXT(GOTOUP2);
:
:
Dstats  ASSIGN: Maqx2=2*(x(6)+1); *** x(6)= N§ de maq. ***
GOTOUP4  ASSIGN: ind2=ind2+1:
:      BRANCH,1:
:      IF, ind2 <= Maqx2+3, PRINT4: ! ** Maqx2 = N§ de DSTATS
:      Else, Print5;          relativos as Maq. **
PRINT4  WRITE, OUT, "(2F6.2)": DAVG(ind2), DMAX(ind2):
:      NEXT(GOTOUP4);
:
Print5  Assign: Tfinal = TNOW;
:      Write, OUT, "(F7.2)": Tfinal:
:      Dispose;
END:

```

ANEXO 5

CÓDIGO FONTE DO EXPERIMENT EM SIMAN IV

```

BEGIN;
PROJECT, FMS, PF;
VARIABLES:      NEnt: Tfinal:
                 NMaxEnt: NPecas:NTal1:NTal2:NTal3:
                 Ind1:Ind2:
                 Maqx2;
ATTRIBUTES:     PT:
                 DEST:
                 OPNUM:
                 TimeIn;
;               Tfila: TFAGV: ttransp;
FILES:          I, OUT, "c:\siman\tese\simOUT.txt",SEQ,FREE;
RESOURCES:      MACH(4):
                 PALLET;
QUEUES:         QMACH1:QMACH2:QMACH3:QMACH4:
                 SaidaQ1:SaidaQ2:SaidaQ3:SaidaQ4:
                 desembQ1:desembQ2:desembQ3:desembQ4:
                 LOADQUEUE:
                 AGVQUEUE:
                 AltQ:PRINTQ;
STATIONS:       1,STA1,14:
                 2,STA2,15:
                 3,STA3,16:
                 4,STA4,17:
                 5,STALOAD,12:
                 6,STAUNLOAD,13:
                 7,STASTAGE,8;
INTERSECTIONS:  1,IntMachine1:
                 2,IntMachine2:
                 3,IntMachine3:
                 4,IntMachine4:
                 5:6:
                 7,IntStage:
                 8,STAGE:
                 9:
                 10,IntUnload:
                 11,IntLoad:
                 12,LOAD:
                 13,UNLOAD:
                 14,MACHINE1:
                 15,MACHINE2:
                 16,MACHINE3:
                 17,MACHINE4:

```

```

18,IntGaragem:
19,Garagem;
LINKS: 1,Link1,1,2,10,5,U:
        2,Link2,2,6,5,5,U:
        3,Link3,6,7,6,5,U:
        4,Link4,7,8,10,5,U: !originalmente ..7,5,U:
        5,Link5,8,9,4,5,U: ! " ..7,5,U:
        6,Link6,9,10,6,5,U:
        7,Link7,10,11,2,5,U:
        8,Link8,11,5,8,5,U:
        9,Link9,5,3,11,5,U:
        10,Link10,3,4,10,5,U:
        11,Link11,4,18,5,5,U:
        12,Link12,7,9,10,5,U:
        13,Link13,10,13,2,5,Spur:
        14,Link14,11,12,2,5,Spur:
        15,Link15,1,14,2,5,Spur:
        16,Link16,2,15,2,5,Spur:
        17,Link17,3,16,2,5,Spur:
        18,Link18,4,17,2,5,Spur:
        19,Link19,5,1,5,5,U:
        20,Link20,18,6,6,5,U:
        21,Link21,19,18,10,5,U:
NETWORKS: 1,AGVPath,1-21:
TRANSPORTERS: AGV,4,NETWORK(AGVPath)-S.80.
              Link(21)-A-ZONE(1):
COUNTERS: 1, N$ de vezes agv parou:
           2, N$ de vezes maq. bloq:
TALLIES:
           1,Tempo de Fluxo PT 1:
           2,Tempo de Fluxo PT 2:
           3,Tempo de Fluxo PT 3:
           4,Tempo de Fluxo PT 4:
           5,Tempo de Fluxo PT 5:
           6,TFluxoTot,:
           7,TempoFilaAGV:
           8,TTranspTotal:
           9,TEspTotal:
DSTATS: NQ(1),Maq #1 Queue Length:
         NQ(2),Maq #2 Queue Length:
         NQ(3),Maq #3 Queue Length:
         NQ(4),Maq #4 Queue Length:
         NR(1),Machine #1 Util:
         NR(2),Machine #2 Util:
         NR(3),Machine #3 Util:
         NR(4),Machine #4 Util:
         NQ(14),AGV QUEUE:
         NT(1),AGV UTIL:
         NR(5),PALLET Util:

:TRACE:
replicate,10,...yes,yes:
END:

```

ANEXO 6

CÓDIGO FONTE DO PROTÓTIPO COM TODAS AS CLASSES,OBJETOS, INTÂNCIAS, FUNÇÕES E MÉTODOS

```

/*****
**      ALL FUNCTIONS ARE SAVED BELOW      **
*****/

/*****
**** FUNCTION: DefParâmAnálise
*****/
MakeFunction( DefParâmAnálise, []).
{
PostInputForm( "Definição de Parâmetros para iniciar o Processo Iterativo de Simulação/Análise".
Máquinas. TamMáxBufferEnt. "Tamanho máximo do Buffer". VAGs.
MaxUtilVAGs. "Máxima Taxa de Utilização dos VAGs". Pallets.
NMinPallets. "Nº Mínimo de Pallets". Pallets. NMaxPallets.
"Nº Máximo de Pallets". Sistema. TCPEst. "Máximo Tempo Estimado para o Ciclo Produtivo" ).
Sistema:TMédCPSimAnt = Sistema:TCPEst;
Pallets:NLimPallets = Pallets:NMaxPallets;
} );

/*****
**** FUNCTION: ParPeçasAnálise
*****/
MakeFunction( ParPeçasAnálise, []).
{
PostMultipleSelection( "Escolha a(s) Peça(s) para mudar os Parâmetros".
Peças:Lista_Atualizada. GetInstanceList( Peças. Peças:ListaPeças ) );
EnumList( Peças:Lista_Atualizada. x.
{
PostMessage( x. " Será atualizada" ).
PostInputForm( "Entre com o Parâmetro da Peça". x. Meta_Prod.
"Produção desejada para essa Peça" ).
} ).
} );

/*****
**** FUNCTION: ResSimPeças
*****/
MakeFunction( ResSimPeças, []).
{
PostMultipleSelection( "Escolha Peças para demonstração dos resultados da Simulação ".
Peças:Lista_Atualizada. GetInstanceList( Peças. Peças:ListaPeças ) );
EnumList( Peças:Lista_Atualizada. x.
{

```

```

PostMessage( x, " Será Mostrada" );
PostInputForm( "Resultados da Simulação para a Peça". x.
  ProdSim. "Produção desta peça". x. TfxSim. " Lead Time Médio para esta Peça" );
} );
} );
SetFunctionComment( ResSimPeças. FormatValue ( "Voce Pode Utilisar Esta Opcao Para Ver e/ou Alterar
Diretamente os Re -
sultados da Simulacao Objetivando Analises do Tipo \" O QUE . SE ?\". " ) );

/*****
**** FUNCTION: Restaura_Diag_Peças
*****/
MakeFunction( Restaura_Diag_Peças. []).
{
PostMessage( "All Parts Operational Objectives Diagnostics Were Reseted" );
ForAll [ x|Peças ]
  ResetValue( x:Diag_de_Objctivos_Operac ).
ResetValue( x:Diag_Tempo_Passagem ).
} );

/*****
**** FUNCTION: Restaura_Diag_Máq
*****/
MakeFunction( Restaura_Diag_Máq. []).
{
PostMessage( "All Machines Local Oper Diag. Global Oper Diag and Mach Action Were Reseted" );
ForAll [ x|Máquinas ]
{
  ResetValue( x:Diag_Oper_Local_Máq ).
  ResetValue( x:Diag_Oper_Global_Máq ).
  ResetValue( x:Ação_na_Máq ).
}
} );

/*****
**** FUNCTION: Par_Oper_para_Simul
*****/
MakeFunction( Par_Oper_para_Simul. []).
{
PostMultipleSelection( "Chose the Operations you Want to See or Change Parameters for Simulation Purposes.".
  Operações:Lista_Atualizada. GetInstanceList( Operações.
    Operações:List_of_Operations ) );
EnumList( Operações:Lista_Atualizada. x.
{
  PostMessage( x, " Will Be Showed" )
PostInputForm( "Enter the Parameters for the Operations".
  x. Best_Time. "Time for this Operation on the Most Apropiated Machine ".
  x. List_of_Possibles_Machines. "Machines where the Operation can be Performed. Decrescent Order." );
} );
} );

/*****
**** FUNCTION: ParCenSim

```

```

*****/
MakeFunction( ParCenSim, [],
{
  Let [x PostMenu( "Escolha o Cenário que deseja Alterar".
    GetSubClassList( Cenários, Cenários:ListCen ) )]
  PostInputForm( "Entre os Parâmetros para a Simulação do Cenário",
    x, NumPeças, "Número de Peças", x, TiposPeças, "Tipos de Peças",
    x, QuantTot, "Quantidade Total de Peças", x, QuantporTipo,
    "Quantidade de Peças por Tipo" );
} );

/*****
**** FUNCTION: ParMáqSim
*****/
MakeFunction( ParMáqSim, [],
{
  PostMultipleSelection( "Escolha a(s) Máquina(s) para Definição de Parâmetros para Simulação",
    Máquinas:Lista_Atualizada, GetInstanceList( Máquinas, Máquinas:ListaMáquinas ) );
  EnumList( Máquinas:Lista_Atualizada, x,
  {
    PostMessage( x, " Será Mostrada" );
    PostInputForm( "Digite os Parâmetros para a Máquina". x,
      ListaOperacoes, "Liste as Operações disponíveis para esta máquina".
      x, Efic_Máq, "Eficiências da Máquina". x, MinUtilMáq,
      "Mínima Utiliz. Desejada", x, MaxUtilMáq, "Máxima Utiliz. Permitida",
      x, TamBufferEnt, " Tamanho do Buffer (Inicial)" );
  } );
} );

/*****
**** FUNCTION: ParVAGPalletBufferSimul
*****/
MakeFunction( ParVAGPalletBufferSimul, [],
{
  PostInputForm( "Entrada de Parâmetros dos VAG's. Pallets e Buffers para a Simulação ",
    Pallets, NPallets, "Digite o Número de Pallets". VAGs, NovoNumVAGs,
    "Digite o Número de VAG's", Máquinas, TamBufferEnt, "Digite o Tamanho do Buffer de Entrada das
Máquinas" );
} );

/*****
**** FUNCTION: Executa_Simul
*****/
MakeFunction( Executa_Simul, [],
{
  AskValue( Global:Exec_Simul_File );
  AskValue( Global:Exec_Simul_Model );
  AskValue( Sistema:NRep );
  Execute( Global:Exec_Simul_File, Global:Exec_Simul_Model );
  PostMessage( "Simulação Terminada" );
  ShowWindow( Análise );
} );

```

```

/*****
**** FUNCTION: Res_Sim_Acessórios
*****/
MakeFunction( Res_Sim_Acessórios, []).
{
PostInputForm( "Resultados da Simulação para os Pallets", Pallets,
UtilMéd, "Taxa de Utilização", Pallets. MáxNumUtil, "Número Máximo Utilizado" );
} );

/*****
**** FUNCTION: Res_Sim_Estrados
*****/
MakeFunction( Res_Sim_Estrados, []).
{
PostInputForm( " Simulation Results for the Pallets", Pallets,
Pallets_Util_Diag, " Pallets Utilization Diag.", Pallets.
Diag_Fila, " Queue Length Diag. ", Pallets. Pallets_LU_Diag,
"Load/Unload Time Diag." );
} );

/*****
**** FUNCTION: Restaura_Diag_Estrados
*****/
MakeFunction( Restaura_Diag_Estrados, []).
{
Global:Selection1 = PostMenu( "All Pallets Diagnostics and Actions will be Reset".
YES. NO );
If ( Global:Selection1 #= YES )
Then {
ResetValue( Pallets:Pallets_LU_Diag );
ResetValue( Pallets:Pallets_Action );
ResetValue( Pallets:Pallets_Diag );
};
} );

/*****
**** FUNCTION: Restaura_Diag_AGVs
*****/
MakeFunction( Restaura_Diag_AGVs, []).
{
Global:Selection1 = PostMenu( "All AGVs Diagnostics and Actions will be Reset".
YES. NO );
If ( Global:Selection1 #= YES )
Then {
ResetValue( VAGs:Speed_Diag );
ResetValue( VAGs:AGV_Action );
ResetValue( VAGs:AGV_Diag );
ResetValue( VAGs:AGV_Global_Diag );
ResetValue( VAGs:AGV_Reliability_Diag );
};
} );

/*****

```


1. The first part of the document is a list of the names of the persons who have been appointed to the various positions of the Board of Directors of the Corporation.

2. The second part of the document is a list of the names of the persons who have been appointed to the various positions of the Board of Directors of the Corporation.

3. The third part of the document is a list of the names of the persons who have been appointed to the various positions of the Board of Directors of the Corporation.

4. The fourth part of the document is a list of the names of the persons who have been appointed to the various positions of the Board of Directors of the Corporation.

5. The fifth part of the document is a list of the names of the persons who have been appointed to the various positions of the Board of Directors of the Corporation.

6. The sixth part of the document is a list of the names of the persons who have been appointed to the various positions of the Board of Directors of the Corporation.

7. The seventh part of the document is a list of the names of the persons who have been appointed to the various positions of the Board of Directors of the Corporation.

8. The eighth part of the document is a list of the names of the persons who have been appointed to the various positions of the Board of Directors of the Corporation.

9. The ninth part of the document is a list of the names of the persons who have been appointed to the various positions of the Board of Directors of the Corporation.

```

**** FUNCTION: Def_Desp_Peças
*****/
MakeFunction( Def_Desp_Peças, [],
{
ShowImage( Text9 );
PostInputForm( "Definição da Política de Despacho de Peças ".
Peças, Pol_Despacho, "Selecione a Política" );
HideImage( Text9 );
} );

/*****
**** FUNCTION: Desp_VAGs
*****/
MakeFunction( Desp_VAGs, [],
{
ShowImage( Text11 );
PostInputForm( "Definição da Política de Despacho de VAGs para a Simulação".
VAGs, Pol_Despacho, "Selecione a Política" );
HideImage( Text11 );
} );

/*****
**** FUNCTION: Seq_Peças
*****/
MakeFunction( Seq_Peças, [],
{
ShowImage( Text10 );
PostInputForm( "Definição da Política de Sequenciamento de Peças".
Peças, Pol_Seq, "Defina a Política de Sequenciamento" );
HideImage( Text10 );
} );

/*****
**** FUNCTION: ParPeçaSim
*****/
MakeFunction( ParPeçaSim, [],
{
PostMultipleSelection( "Alteração de Características de Peças para Simulação".
Peças:Lista_Atualizada,.GetInstanceList( Peças, Peças:ListaPeças ) );
EnumList( Peças:Lista_Atualizada, x.
{
PostMessage( x, " Será Mostrada " );
PostInputForm( "Atualize os Parâmetros para a Peça". x, ListaOper.
"Liste as Operações que esta peça será submetida". x.
Tempo_Processamento, "Liste os tempos de processamento para as operações acima" );
While (( LengthList( x:ListaOper ) < 10 ))
AppendToList( x:ListaOper, 0 );
While (( LengthList( x:Tempo_Processamento )
< 10 ))
AppendToList( x:Tempo_Processamento, 0 );
} );
} );

```



```

SetValue( VAGs:UtilTotal, 0 );
SetValue( Pallets:UtilTotal, 0 );
SetValue( Sistema:TCPSimTotal, 0 );
ForAll [ x|Máquinas ]
{
    SetValue( x:TamFilaTotal, 0 );
    SetValue( x:UtilMáqTotal, 0 );
};
Let [Cen SelectSubClass( Cenários, x, x #= Cenários:CenSim )]
EnumList( Cen:TiposPeças, tp,
{
    Let [Pc Peça # tp]
    {
        SetValue( Pc:TFxSimTotal, 0 );
        SetValue( Pc:ProdSimTotal, 0 );
    };
});
For Rep From 1 To Sistema:NRep
Do {
    Let [Cen SelectSubClass( Cenários, x, x #= Cenários:CenSim )]
    EnumList( Cen:TiposPeças, tp,
    {
        Let [Pc Peça # tp]
        {
            SetValue( Pc:TFxSim, ReadWord( ) );
            Pc:TFxSimTotal = Pc:TFxSimTotal + Pc:TFxSim;
            SetValue( Pc:ProdSim, ReadWord( ) );
            Pc:ProdSimTotal = Pc:ProdSimTotal + Pc:ProdSim;
        };
    });
    SetValue( Sistema:TFxPSim, ReadWord( ) );
    Sistema:TFxPSimTotal = Sistema:TFxPSimTotal + Sistema:TFxPSim;
    SetValue( Sistema:TPPSim, ReadWord( ) );
    SetValue( Sistema:TFiVAG, ReadWord( ) );
    SetValue( Sistema:ObsTFiVAG, ReadWord( ) );
    Sistema:TFiVAGCalc = ( Sistema:TFiVAG * Sistema:ObsTFiVAG )
        / Sistema:TPPSim;
    Sistema:TFiVAGTotal = Sistema:TFiVAGTotal + Sistema:TFiVAGCalc;
    SetValue( Sistema:TTransp, ReadWord( ) );
    SetValue( Sistema:ObsTTransp, ReadWord( ) );
    Sistema:TTranspCalc = ( Sistema:TTransp * Sistema:ObsTTransp )
        / Sistema:TPPSim;
    Sistema:TTranspTotal = Sistema:TTranspTotal + Sistema:TTranspCalc;
    SetValue( Sistema:TEsp, ReadWord( ) );
    SetValue( Sistema:ObsTEsp, ReadWord( ) );
    Sistema:TEspCalc = ( Sistema:TEsp * Sistema:ObsTEsp )
        / Sistema:TPPSim;
    Sistema:TEspTotal = Sistema:TEspTotal + Sistema:TEspCalc;
    ForAll [ x|Máquinas ]
    {
        SetValue( x:TamFila, ReadWord( ) );
        x:TamFilaTotal = x:TamFilaTotal + x:TamFila;
        SetValue( x:Máx_Tam_Fila_na_Simul,

```

```

        ReadWord( );
    };
    ForAll [ x|Máquinas ]
    {
        SetValue( x:UtilMáq, ReadWord( ) );
        x:UtilMáqTotal = x:UtilMáqTotal + x:UtilMáq;
        SetValue( x:MáxNumUtil, ReadWord( ) );
    };
    SetValue( VAGs:TamFila, ReadWord( ) );
    VAGs:TamFilaTotal = VAGs:TamFilaTotal + VAGs:TamFila;
    SetValue( VAGs:Máx_Tam_Fila_na_Simul, ReadWord( ) );
    SetValue( VAGs:Util, ReadWord( ) );
    VAGs:UtilTotal = VAGs:UtilTotal + VAGs:Util;
    SetValue( VAGs:MáxNumUtil, ReadWord( ) );
    SetValue( Pallets:Util, ReadWord( ) );
    Pallets:UtilTotal = Pallets:UtilTotal + Pallets:Util;
    SetValue( Pallets:MáxNumUtil, ReadWord( ) );
    SetValue( Sistema:TCPSim, ReadWord( ) );
    Sistema:TCPSimTotal = Sistema:TCPSimTotal + Sistema:TCPSim;
};
CloseReadFile( );
Let [Cen SelectSubClass( Cenários, x, x #= Cenários:CenSim )]
EnumList( Cen:TiposPeças, tp,
{
    Let [Pc Peça # tp]
    {
        Pc:TMédFxSim = Pc:TFxSimTotal / Sistema:NRep;
        Pc:ProdMédSim = Pc:ProdSimTotal / Sistema:NRep;
    };
});
Sistema:TMédFxPSim = Sistema:TFxPSimTotal / Sistema:NRep;
Sistema:TMédFiVAG = Sistema:TFiVAGTotal / Sistema:NRep;
Sistema:TMédTransp = Sistema:TTranspTotal / Sistema:NRep;
Sistema:TMédEsp = Sistema:TEspTotal / Sistema:NRep;
ForAll [ x|Máquinas ]
{
    x:TamMédFila = x:TamFilaTotal / Sistema:NRep;
    x:UtilMédMáq = x:UtilMáqTotal / Sistema:NRep;
};
AtualListaMáqSobreUtil( );
VAGs:TamMédFila = VAGs:TamFilaTotal / Sistema:NRep;
VAGs:UtilMéd = VAGs:UtilTotal / Sistema:NRep;
Pallets:UtilMéd = Pallets:UtilTotal / Sistema:NRep;
Sistema:TMédCPSim = Sistema:TCPSimTotal / Sistema:NRep;
PostMessage( "Todos os Resultados da Simulação Foram Atualizados" );
});

/*****
**** FUNCTION: ResSimMáq
*****/
MakeFunction( ResSimMáq3, [],
{
    ClearTranscriptImage( Transcript1 );

```



```

WriteLine( a:NumPeças - 1 );
WriteLine( Máquinas:Num_de_Máquinas - 1 );
Let [a SelectSubClass( Cenários, x, x #= Cenários:CenSim )]
  WriteLine( a:QuantTot );
Let [a SelectSubClass( Cenários, x, x #= Cenários:CenSim )]
  EnumList( a:QuantporTipo, QpT, WriteLine( QpT ) );
Let [a SelectSubClass( Cenários, x, x #= Cenários:CenSim )]
  EnumList( a:TiposPeças, tp,
    {
      Let [pc Peça # tp]
        EnumList( pc:ListaOper, LO, WriteLine( LO ) );
    } );
Let [a SelectSubClass( Cenários, x, x #= Cenários:CenSim )]
  EnumList( a:TiposPeças, tp,
    {
      Let [pc Peça # tp]
        EnumList( pc:Tempo_Processamento, LO,
          WriteLine( LO ) );
    } );
ForAll [ x|Máquinas ]
  EnumList( x:Efic_Máq, em, WriteLine( em ) );
CloseWriteFile( );
PostMessage( "Parâmetros Salvos" );
} );

/***** FUNCTION: IniciaDiagSistema
*****/
MakeFunction( IniciaDiagSistema, [],
{
  PostMessage( "Iniciando Análise dos Resultados da Simulação" );
  ResetValue( Sistema:DiagSistema );
  MakeGoal( Goal # Sim, KnownValue?( Sistema:DiagSistema ) );
  BackwardChain( Goal # Sim, Sistema:ListRegrDiagOtimzSistema );
  DeleteGoal( Goal # Sim );
  If ( Sistema:Menu #= Menu6 )
    Then ShowWindow( ResultSimul )
    Else PopupMenu( Sistema:Menu );
} );

/***** FUNCTION: SalvaParSimAnálise
*****/
MakeFunction( SalvaParSimAnálise, [],
{
  OpenWriteFile( Global:ArqParVAGs );
  EnumList( VAGs:ListaEstadoVAGs, IT, WriteLine( IT ) );
  CloseWriteFile( );
  OpenWriteFile( Global:ArqParSim );
  WriteLine( Máquinas:TamBufferEnt );
  WriteLine( Máquinas:TamBufferSaida );
  WriteLine( Pallets:NPallets );
  WriteLine( Operações:Num_Oper - 1 );

```

```

Let [a SelectSubClass( Cenários, x, x #= Cenários:CenSim )]
  WriteLine( a:NumPeças - 1 );
WriteLine( Máquinas:Num_de_Máquinas - 1 );
Let [a SelectSubClass( Cenários, x, x #= Cenários:CenSim )]
  WriteLine( a:QuantTot );
Let [a SelectSubClass( Cenários, x, x #= Cenários:CenSim )]
  EnumList( a:QuantporTipo, QpT, WriteLine( QpT ) );
Let [a SelectSubClass( Cenários, x, x #= Cenários:CenSim )]
  EnumList( a:TiposPeças, tp,
    {
      Let [pc Peça # tp]
        EnumList( pc:ListaOper, LO, WriteLine( LO ) );
    } );
Let [a SelectSubClass( Cenários, x, x #= Cenários:CenSim )]
  EnumList( a:TiposPeças, tp,
    {
      Let [pc Peça # tp]
        EnumList( pc:Tempo_Processamento, LO,
          WriteLine( LO ) );
    } );
ForAll [ x|Máquinas ]
  EnumList( x:Efic_Máq, em, WriteLine( em ) );
CloseWriteFile( );
} );

```

```

/****
/**      ALL CLASSES ARE SAVED BELOW      **/
****

```

```

/****
**** CLASS: Máquinas
****

```

```
MakeClass( Máquinas, Root );
```

```

/**** METHOD: Diag_Util_Máq ****
MakeMethod( Máquinas, Diag_Util_Máq, []).
If ( Self:UtilMédMáq < Self:MinUtilMáq )
  Then ( Self:Diag_Util_Máq = Baixa )
Else If ( Self:UtilMédMáq <= Self:MaxUtilMáq )
  Then ( Self:Diag_Util_Máq = Normal )
Else Self:Diag_Util_Máq = Alta );

```

```

/**** METHOD: Diag_Fila_Máq ****
MakeMethod( Máquinas, Diag_Fila_Máq, []).
If ( Self:TamMédFila < 1.0 )
  Then ( Self:Diag_Fila = Curta )
Else If ( Self:TamMédFila <= Self:TamBufferEnt )
  Then Self:Diag_Fila = Normal );
MakeSlot( Máquinas:UtilMáq );
MakeSlot( Máquinas:TamFila );

```



```

MakeSlot( Máquinas:MinUtilMáq );
MakeSlot( Máquinas:MaxUtilMáq );
MakeSlot( Máquinas:Diag_Util_Máq );
SetSlotOption( Máquinas:Diag_Util_Máq, ALLOWABLE_VALUES, Alta, Normal, Baixa );
MakeSlot( Máquinas:Diag_Fila );
SetSlotOption( Máquinas:Diag_Fila, ALLOWABLE_VALUES, Curta, Normal, Longa );
MakeSlot( Máquinas:ListaMáquinas );
SetSlotOption( Máquinas:ListaMáquinas, MULTIPLE );
SetValue( Máquinas:ListaMáquinas, Máq1, Máq2, Máq3, Máq4 );
MakeSlot( Máquinas:DesvioCompFila );
MakeSlot( Máquinas:DesvioUtil );
MakeSlot( Máquinas:TamBufferEnt );
SetSlotOption( Máquinas:TamBufferEnt, ALLOWABLE_VALUES, 2, 3, 4, 5, 6 );
Máquinas:TamBufferEnt = 5;
MakeSlot( Máquinas:MinNumUtil );
MakeSlot( Máquinas:MáxNumUtil );
MakeSlot( Máquinas:Num_de_Máquinas );
Máquinas:Num_de_Máquinas = 4;
SetSlotOption( Máquinas:Num_de_Máquinas, AFTER_CHANGE, DiagUtilRelMáq );
MakeSlot( Máquinas:Máx_Tam_Fila_na_Simul );
MakeSlot( Máquinas:Min_Tam_Fila_na_Simul );
MakeSlot( Máquinas:DiagConfMáq );
SetSlotOption( Máquinas:DiagConfMáq, ALLOWABLE_VALUES, OK, Ruim );
Máquinas:DiagConfMáq = OK;
MakeSlot( Máquinas:ListaOperacoes );
SetSlotOption( Máquinas:ListaOperacoes, ALLOWABLE_VALUES, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 );
ClearList( Máquinas:ListaOperacoes );
MakeSlot( Máquinas:TamBufferSaida );
SetSlotOption( Máquinas:TamBufferSaida, NO_AUTO_ASK, TRUE );
Máquinas:TamBufferSaida = 1;
MakeSlot( Máquinas:TamFilaTotal );
MakeSlot( Máquinas:UtilMáqTotal );
MakeSlot( Máquinas:TamMédFila );
SetSlotOption( Máquinas:TamMédFila, AFTER_CHANGE, Diag_Fila_Máq );
MakeSlot( Máquinas:UtilMédMáq );
SetSlotOption( Máquinas:UtilMédMáq, AFTER_CHANGE, Diag_Util_Máq );
MakeSlot( Máquinas:ListaMáqSobreUtil );
SetSlotOption( Máquinas:ListaMáqSobreUtil, ALLOWABLE_VALUES, Máq1, Máq2, Máq3, Máq4 );
ClearList( Máquinas:ListaMáqSobreUtil );
MakeSlot( Máquinas:TamMáxBufferEnt );
Máquinas:TamMáxBufferEnt = 6;
MakeSlot( Máquinas:Lista_Atualizada );
SetSlotOption( Máquinas:Lista_Atualizada, MULTIPLE );
SetValue( Máquinas:Lista_Atualizada, Máq1 );
MakeSlot( Máquinas:Efic_Máq );
SetSlotOption( Máquinas:Efic_Máq, MINIMUM_VALUE, 0 );
SetSlotOption( Máquinas:Efic_Máq, MAXIMUM_VALUE, 1 );
ClearList( Máquinas:Efic_Máq );

```

```

/*****

```

```

**** CLASS: Peças

```

```

*****/

```

```

MakeClass( Peças, Root );

```

```

/***** METHOD: Diag_de_Produção *****/
MakeMethod( Peças, Diag_de_Produção, [],
  If ( Self:ProdSim < Self:Meta_Prod * .90 )
    Then ( Self:Diag_de_Objctivos_Operac = Não_Alcançado )
    Else Self:Diag_de_Objctivos_Operac = Alcançado );

/***** METHOD: Diag_Tempo_Passagem *****/
MakeMethod( Peças, Diag_Tempo_Passagem, [],
  If ( Self:TFxSim > Self:Tempo_Esperado_Passagem * 0.90 )
    Then ( Self:Diag_Tempo_Passagem = Alta )
    Else Self:Diag_Tempo_Passagem = Aceitável );

/***** METHOD: NewTempoEspPassag *****/
MakeMethod( Peças, NewTempoEspPassag, [],
  {
    Self:Tempo_Acumulado_Processamento = Sum( Self:Tempo_Processamento );
    Self:Tempo_Esperado_Passagem = 3 * Self:Tempo_Acumulado_Processamento;
  } );
MakeSlot( Peças:Meta_Prod );
SetSlotOption( Peças:Meta_Prod, AFTER_CHANGE, Diag_de_Produção );
MakeSlot( Peças:Tempo_Processamento );
SetSlotOption( Peças:Tempo_Processamento, MULTIPLE );
SetValue( Peças:Tempo_Processamento, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 );
MakeSlot( Peças:ListaOper );
SetSlotOption( Peças:ListaOper, MULTIPLE );
SetSlotOption( Peças:ListaOper, ALLOWABLE_VALUES, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 );
SetValue( Peças:ListaOper, 2, 1, 3, 5, 3, 2 );
MakeSlot( Peças:ProdSim );
MakeSlot( Peças:Tempo_Esperado_Passagem );
SetSlotOption( Peças:Tempo_Esperado_Passagem, WHEN_ACCESS, NewTempoEspPassag );
MakeSlot( Peças:Num_Oper );
MakeSlot( Peças:TFxSim );
MakeSlot( Peças:Diag_de_Objctivos_Operac );
SetSlotOption( Peças:Diag_de_Objctivos_Operac, ALLOWABLE_VALUES, Alcançado, Não_Alcançado );
MakeSlot( Peças:Tempo_Acumulado_Processamento );
MakeSlot( Peças:Diag_Tempo_Passagem );
SetSlotOption( Peças:Diag_Tempo_Passagem, ALLOWABLE_VALUES, Aceitável, Alto );
MakeSlot( Peças:ListaPeças );
SetValue( Peças:ListaPeças, Peça1, Peça2, Peça3, Peça4, Peça5, Peça6, Peça7, Peça8, Peça9, Peça10, Peça11,
Peça12, Peça13, Peça14, Peça15, Peça16, Peça17, Peça18, Peça19, Peça20 );
MakeSlot( Peças:Pol_Despacho );
SetSlotOption( Peças:Pol_Despacho, ALLOWABLE_VALUES, RAN, MTxP, MNV, MTP );
Peças:Pol_Despacho = MTP;
MakeSlot( Peças:Pol_Seq );
SetSlotOption( Peças:Pol_Seq, ALLOWABLE_VALUES, FIFO, MTP, MTPR, MTPPTP );
Peças:Pol_Seq = FIFO;
MakeSlot( Peças:TempoMédioTransp );
MakeSlot( Peças:TFxSimTotal );
MakeSlot( Peças:ProdSimTotal );
MakeSlot( Peças:TMédFxSim );
SetSlotOption( Peças:TMédFxSim, AFTER_CHANGE, Diag_Tempo_Passagem );
MakeSlot( Peças:ProdMédSim );

```

```
SetSlotOption( Peças:ProdMédSim, AFTER_CHANGE, Diag_de_Produção );
MakeSlot( Peças:Lista_Atualizada );
SetValue( Peças:Lista_Atualizada, Peça2 );
```

```
/******
**** CLASS: Transportadores
*****/
```

```
MakeClass( Transportadores, Root );
MakeSlot( Transportadores:List_of_Transporters );
SetValue( Transportadores:List_of_Transporters, VAGs );
MakeSlot( Transportadores:Lista_Atualizada );
SetValue( Transportadores:Lista_Atualizada, VAGs );
```

```
/******
**** CLASS: Acessórios
*****/
```

```
MakeClass( Acessórios, Root );
MakeSlot( Acessórios:List_of_Accessories );
SetValue( Acessórios:List_of_Accessories, Pallets );
MakeSlot( Acessórios:Lista_Atualizada );
SetValue( Acessórios:Lista_Atualizada, Pallets );
```

```
/******
**** CLASS: Operações
**** This class should contain all types of operations that
**** could be performed by the FMS.
*****/
```

```
MakeClass( Operações, Root );
SetClassComment( Operações, "This class should contain all types of operations that
could be performed by the FMS. " );
MakeSlot( Operações:Best_Time );
MakeSlot( Operações:List_of_Possible_Machines );
ClearList( Operações:List_of_Possible_Machines );
MakeSlot( Operações:List_of_Operations );
SetValue( Operações:List_of_Operations, Operação1, Operação2, Operação3, Operação4, Operação5 );
MakeSlot( Operações:Lista_Atualizada );
SetValue( Operações:Lista_Atualizada, Operação1 );
MakeSlot( Operações:Num_Oper );
Operações:Num_Oper = 10;
```

```
/******
**** CLASS: Cenários
*****/
```

```
MakeClass( Cenários, Root );
```

```
/****** METHOD: AtualizMetaProd *****/
MakeMethod( Cenários, AtualizMetaProd, []).
{
  For i From 1 To Self:NumPeças
```

```

    Do {
    MakeSlot( Global:QP );
    MakeSlot( Global:PP );
    SetValue( Global:QP, GetNthElem( Self:QuantporTipo, i ) );
    SetValue( Global:PP, GetNthElem( Self:TiposPeças, i ) );
    Let [Pc Peça # Global:PP]
        SetValue( Pc:Meta_Prod, Global:QP );
    };
});
MakeSlot( Cenários:CenSim );
Cenários:CenSim = Cenário1;
MakeSlot( Cenários:ListCen );
SetValue( Cenários:ListCen, Cenário1, Cenário2, Cenário3 );
MakeSlot( Cenários:NumPeças );
MakeSlot( Cenários:TiposPeças );
SetSlotOption( Cenários:TiposPeças, ALLOWABLE_VALUES, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20 );
ClearList( Cenários:TiposPeças );
MakeSlot( Cenários:QuantporTipo );
ClearList( Cenários:QuantporTipo );
SetSlotOption( Cenários:QuantporTipo, AFTER_CHANGE, AtualizMetaProd );
MakeSlot( Cenários:QuantTot );

/*****
**** CLASS: Cenário1
*****/

MakeClass( Cenário1, Cenários );
Cenário1:NumPeças = 5;
SetValue( Cenário1:TiposPeças, 2, 4, 13, 17, 18 );
Cenário1:QuantTot = 108;
SetValue( Cenário1:QuantporTipo, 24, 18, 12, 24, 30 );

/*****
**** CLASS: Cenário2
*****/

MakeClass( Cenário2, Cenários );
Cenário2:NumPeças = 6;
SetValue( Cenário2:TiposPeças, 1, 2, 3, 4, 5 );
Cenário2:QuantTot = 20;
SetValue( Cenário2:QuantporTipo, 3, 3, 3, 3, 3, 4 );

/*****
**** CLASS: Cenário3
*****/

MakeClass( Cenário3, Cenários );
Cenário3:NumPeças = 6;
SetValue( Cenário3:TiposPeças, 4, 5, 6, 7, 8, 9 );
Cenário3:QuantTot = 50;
SetValue( Cenário3:QuantporTipo, 10, 10, 10, 10, 5, 5 );

```

```

/*****
**** CLASS: Sistema
*****/

```

```
MakeClass( Sistema, Root );
```

```

/***** METHOD: AtivaMenu *****/

```

```
MakeMethod( Sistema, AtivaMenu, []).
```

```
    PopupMenu( Sistema.Menu );
```

```
    MakeSlot( Sistema:TFxPSim );
```

```
    SetSlotComment( Sistema:TFxPSim, "TMFPSim = Tempo Médio Simulado do Fluxo de Peças" );
```

```
    Sistema:TFxPSim = 140.38;
```

```
    MakeSlot( Sistema:TPPSim );
```

```
    SetSlotComment( Sistema:TPPSim, "TPPSim = Total Simulado de Peças Prod. " );
```

```
    Sistema:TPPSim = 108.00;
```

```
    MakeSlot( Sistema:TCPSim );
```

```
    SetSlotComment( Sistema:TCPSim, "TCPSim = Tempo Simulado do Ciclo de Produção" );
```

```
    Sistema:TCPSim = 1952.13;
```

```
    MakeSlot( Sistema:TCPEst );
```

```
    Sistema:TCPEst = 5000;
```

```
    MakeSlot( Sistema:TPPEst );
```

```
    MakeSlot( Sistema:TMédFxPEst );
```

```
    MakeSlot( Sistema:DiagTCP );
```

```
    SetSlotOption( Sistema:DiagTCP, ALLOWABLE_VALUES, "TCP Aceitável", "TCP Não Aceitável" );
```

```
    MakeSlot( Sistema:DiagUtilMáq );
```

```
    SetSlotOption( Sistema:DiagUtilMáq, ALLOWABLE_VALUES, Aceitável, "Não Aceitável" );
```

```
    MakeSlot( Sistema:DiagUtilVAGs );
```

```
    SetSlotOption( Sistema:DiagUtilVAGs, ALLOWABLE_VALUES, "Baixa Utilização", "Utilização Normal", "Alta Utilização", Normal, Alta, Utilização, Normal, Alta, Utilização );
```

```
    MakeSlot( Sistema:DiagTMFP );
```

```
    SetSlotOption( Sistema:DiagTMFP, ALLOWABLE_VALUES, "TMF Aceitável", "TMF Não Aceitável" );
```

```
    MakeSlot( Sistema:NRep );
```

```
    SetSlotOption( Sistema:NRep, MINIMUM_VALUE, 1 );
```

```
    SetSlotOption( Sistema:NRep, MAXIMUM_VALUE, 10 );
```

```
    Sistema:NRep = 5;
```

```
    SetSlotOption( Sistema:NRep, PROMPT, "Nº de Replicações" );
```

```
    MakeSlot( Sistema:TFxPSimTotal );
```

```
    SetSlotOption( Sistema:TFxPSimTotal, MINIMUM_VALUE, 0 );
```

```
    Sistema:TFxPSimTotal = 700.38;
```

```
    MakeSlot( Sistema:TFiVAG );
```

```
    Sistema:TFiVAG = 9.01;
```

```
    MakeSlot( Sistema:ObsTFiVAG );
```

```
    Sistema:ObsTFiVAG = 564.00;
```

```
    MakeSlot( Sistema:TFiVAGCalc );
```

```
    Sistema:TFiVAGCalc = 47.05222222222222;
```

```
    MakeSlot( Sistema:TFiVAGTotal );
```

```
    SetSlotOption( Sistema:TFiVAGTotal, MINIMUM_VALUE, 0 );
```

```
    Sistema:TFiVAGTotal = 0;
```

```
    MakeSlot( Sistema:TTransp );
```

```
    Sistema:TTransp = 4.08;
```

```
    MakeSlot( Sistema:ObsTTransp );
```

```
    Sistema:ObsTTransp = 456.00;
```

```
    MakeSlot( Sistema:TTranspCalc );
```

```

Sistema:TTranspCalc = 17.22666666666667;
MakeSlot( Sistema:TTranspTotal );
SetSlotOption( Sistema:TTranspTotal, MINIMUM_VALUE, 0 );
Sistema:TTranspTotal = 0;
MakeSlot( Sistema:TEsp );
Sistema:TEsp = 11.12;
MakeSlot( Sistema:ObsTEsp );
Sistema:ObsTEsp = 594.00;
MakeSlot( Sistema:TEspCalc );
Sistema:TEspCalc = 61.16;
MakeSlot( Sistema:TEspTotal );
SetSlotOption( Sistema:TEspTotal, MINIMUM_VALUE, 0 );
Sistema:TEspTotal = 0;
MakeSlot( Sistema:TCPSimTotal );
SetSlotOption( Sistema:TCPSimTotal, MINIMUM_VALUE, 0 );
Sistema:TCPSimTotal = 9706.56;
MakeSlot( Sistema:TMédCPSim );
Sistema:TMédCPSim = 1941.312;
MakeSlot( Sistema:TMédFiVAG );
Sistema:TMédFiVAG = 40.03930555555556;
MakeSlot( Sistema:TMédTransp );
Sistema:TMédTransp = 17.1012962962963;
MakeSlot( Sistema:TMédEsp );
Sistema:TMédEsp = 68.42;
MakeSlot( Sistema:TMédFxPSim );
Sistema:TMédFxPSim = 140.076;
MakeSlot( Sistema:TMédCPSimAnt );
Sistema:TMédCPSimAnt = 1941.312;
MakeSlot( Sistema:DiagSistema );
Sistema:DiagSistema = "R3 - Os resultados da simulação mostraram melhoras no desempenho do sistema. ";
MakeSlot( Sistema:ListRegrDiagOtimzSistema );
SetValue( Sistema:ListRegrDiagOtimzSistema, ROTm1, ROTm2, ROTm3, ROTm4, ROTm5, ROTm6, ROTm7,
ROTm8, ROTm9, ROTm10 );
MakeSlot( Sistema:Menu );
Sistema:Menu = SimÑSim3;

```

```

/*****
/**      ALL INSTANCES ARE SAVED BELOW      **/
*****/

```

```

MakeSlot( Global:Exec_Simul_File );
Global:Exec_Simul_File = "c:\kappa\fms\fmisc14";
MakeSlot( Global:ArqParSim );
Global:ArqParSim = "c:\kappa\fms\dadkap13.txt";
MakeSlot( Global:Sim_Res_File );
Global:Sim_Res_File = "c:\siman\tese\simout.txt";
MakeSlot( Global:QP );
Global:QP = 30;
MakeSlot( Global:PP );
Global:PP = 18;
MakeSlot( Global:Exec_Simul_Model );
Global:Exec_Simul_Model = t140;
MakeSlot( Global:ArqParVAGs );

```

Global:ArqParVAGs = "c:\kappa\fms\dadosit.txt";

```

/*****
**** INSTANCE: Máq1
*****/

```

```

MakeInstance( Máq1, Máquinas );
Máq1:MinUtilMáq = .75;
Máq1:MaxUtilMáq = .90;
Máq1:Diag_Util_Máq = Normal;
Máq1:Diag_Fila = Curta;
Máq1:DesvioCompFila = .69;
Máq1:TamBufferEnt = 2;
Máq1:DesvioUtil = .23;
Máq1:MinNumUtil = .00;
Máq1:MáxNumUtil = 1.00;
Máq1:Min_Tam_Fila_na_Simul = .00;
Máq1:Máx_Tam_Fila_na_Simul = 3.00;
Máq1:DiagConfMáq = OK;
Máq1:UtilMáq = .86;
Máq1:TamFila = .47;
SetValue( Máq1:ListaOperacoes, 2, 4, 5, 6, 8, 10 );
Máq1:TamBufferSaida = 1;
Máq1:TamFilaTotal = 2.58;
Máq1:UtilMáqTotal = 4.41;
Máq1:TamMédFila = 0.516;
Máq1:UtilMédMáq = 0.882;
SetValue( Máq1:Efic_Máq, 0, 1, .75, 0, 0, .75, 1, .75, 1, 0 );

```

```

/*****
**** INSTANCE: Peça1
*****/

```

```

MakeInstance( Peça1, Peças );
SetValue( Peça1:ListaOper, 5, 4, 9, 2, 4, 0, 0, 0, 0, 0 );
SetValue( Peça1:Tempo_Processamento, 6, 14, 11, 7, 11, 0, 0, 0, 0, 0 );
Peça1:TFxSim = 289.52;
Peça1:Tempo_Acumulado_Processamento = 49;
Peça1:Tempo_Esperado_Passagem = 147;
Peça1:Diag_Tempo_Passagem = Alta;
Peça1:ProdSim = 150.00;
Peça1:TFxSimTotal = 1343.74;
Peça1:ProdSimTotal = 342;
Peça1:TMédFxSim = 268.748;
Peça1:ProdMédSim = 68.4;

```

```

/*****
**** INSTANCE: Máq2
*****/

```

```

MakeInstance( Máq2, Máquinas );
Máq2:MinUtilMáq = .75;
Máq2:MaxUtilMáq = .90;
Máq2:UtilMáq = .86;
Máq2:TamFila = .41;

```

```

Máq2:Diag_Util_Máq = Normal;
Máq2:Diag_Fila = Curta;
Máq2:DesvioCompFila = .65;
Máq2:TamBufferEnt = 2;
Máq2:DesvioUtil = .30;
Máq2:MinNumUtil = .00;
Máq2:MáxNumUtil = 1.00;
Máq2:Min_Tam_Fila_na_Simul = .00;
Máq2:Máx_Tam_Fila_na_Simul = 3.00;
SetValue( Máq2:ListaOperacoes, 2, 3, 6, 7, 8, 9 );
Máq2:TamFilaTotal = 2.45;
Máq2:UtilMáqTotal = 4.37;
Máq2:TamMédFila = 0.49;
Máq2:UtilMédMáq = 0.874;
SetValue( Máq2:Efic_Máq, 0, 1, .75, 0, 0, .75, 1, .75, 1, 0 );

```

```

/*****

```

```

**** INSTANCE: Máq3

```

```

*****/

```

```

MakeInstance( Máq3, Máquinas );
Máq3:MinUtilMáq = .75;
Máq3:MaxUtilMáq = .90;
Máq3:UtilMáq = .76;
Máq3:TamFila = .63;
Máq3:DesvioCompFila = .41;
Máq3:TamBufferEnt = 2;
Máq3:DesvioUtil = .49;
Máq3:MinNumUtil = .00;
Máq3:MáxNumUtil = 1.00;
Máq3:Diag_Fila = Curta;
Máq3:Min_Tam_Fila_na_Simul = .00;
Máq3:Máx_Tam_Fila_na_Simul = 4.00;
Máq3:Diag_Util_Máq = Baixa;
SetValue( Máq3:ListaOperacoes, 1, 3, 4, 7, 10 );
Máq3:TamFilaTotal = 2.59;
Máq3:UtilMáqTotal = 3.72;
Máq3:TamMédFila = 0.518;
Máq3:UtilMédMáq = 0.744;
SetValue( Máq3:Efic_Máq, .75, 0, 1, 1, 0, 0, .75, 0, 0, .75 );

```

```

/*****

```

```

**** INSTANCE: Máq4

```

```

*****/

```

```

MakeInstance( Máq4, Máquinas );
Máq4:MinUtilMáq = .75;
Máq4:MaxUtilMáq = .90;
Máq4:UtilMáq = .87;
Máq4:TamFila = .86;
Máq4:Diag_Util_Máq = Normal;
Máq4:Diag_Fila = Curta;
Máq4:DesvioCompFila = .47;
Máq4:TamBufferEnt = 2;
Máq4:DesvioUtil = .43;

```



```

Máq4:MinNumUtil = .00;
Máq4:MáxNumUtil = 1.00;
Máq4:Min_Tam_Fila_na_Simul = .00;
Máq4:Máx_Tam_Fila_na_Simul = 4.00;
SetValue( Máq4:ListaOperacoes, 1, 5, 9 );
Máq4:TamFilaTotal = 4.26;
Máq4:UtilMáqTotal = 4.26;
Máq4:TamMédFila = 0.852;
Máq4:UtilMédMáq = 0.852;
SetValue( Máq4:Efic_Máq, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 );

/*****
**** INSTANCE: Pallets
*****/
MakeInstance( Pallets, Acessórios );

/***** METHOD: Pallets_Queue_Diag *****/
MakeMethod( Pallets, Pallets_Queue_Diag, []).
If ( Self:TamMédFila < 1.0 )
Then ( Self:Diag_Fila = Curta )
Else If ( Self:TamMédFila <= Self:TamBuffer )
Then ( Self:Diag_Fila = Normal )
Else Self:Diag_Fila = Longa );

/***** METHOD: Pallets_Util_Diag *****/
MakeMethod( Pallets, Pallets_Util_Diag, []).
If ( Self:UtilMéd / Self:NPallets < Self:Min_Allow_Util )
Then ( Self:Pallets_Util_Diag = Baixa )
Else If ( Self:UtilMéd / Self:NPallets <= Self:MaxUtilPallets )
Then ( Self:Pallets_Util_Diag = Normal )
Else Self:Pallets_Util_Diag = Alta );

/***** METHOD: DefStep *****/
MakeMethod( Pallets, DefStep, []).
{
If ( Máquinas:TamBufferEnt <= 3 )
Then ( Pallets:Step = 2 )
Else Pallets:Step = 4;
});
MakeSlot( Pallets:TamMédFila );
SetSlotComment( Pallets:TamMédFila, "Show the average number of elements in the Pallet's
Queue." );
Pallets:TamMédFila = .00;
SetSlotOption( Pallets:TamMédFila, AFTER_CHANGE, Pallets_Queue_Diag );
MakeSlot( Pallets:Min_Tam_Fila_na_Simul );
Pallets:Min_Tam_Fila_na_Simul = .00;
MakeSlot( Pallets:Máx_Tam_Fila_na_Simul );
Pallets:Máx_Tam_Fila_na_Simul = .00;
MakeSlot( Pallets:Util );
Pallets:Util = 7.71;
MakeSlot( Pallets:MinNumUtil );
Pallets:MinNumUtil = .00;
MakeSlot( Pallets:MáxNumUtil );

```

```

Pallets:MaxNumUtil = 8.00;
MakeSlot( Pallets:NPallets );
Pallets:NPallets = 16;
MakeSlot( Pallets:Diag_Fila );
SetSlotOption( Pallets:Diag_Fila, ALLOWABLE_VALUES, Curta, Normal, Longa );
Pallets:Diag_Fila = Curta;
MakeSlot( Pallets:TamBuffer );
Pallets:TamBuffer = 5.0;
SetSlotOption( Pallets:TamBuffer, AFTER_CHANGE, Pallets_Queue_Diag );
MakeSlot( Pallets:Min_Allow_Util );
SetSlotOption( Pallets:Min_Allow_Util, MINIMUM_VALUE, 0.0 );
SetSlotOption( Pallets:Min_Allow_Util, MAXIMUM_VALUE, 1.0 );
Pallets:Min_Allow_Util = .65;
SetSlotOption( Pallets:Min_Allow_Util, AFTER_CHANGE, Pallets_Util_Diag );
MakeSlot( Pallets:MaxUtilPallets );
SetSlotOption( Pallets:MaxUtilPallets, MINIMUM_VALUE, 0.0 );
SetSlotOption( Pallets:MaxUtilPallets, MAXIMUM_VALUE, 1.0 );
Pallets:MaxUtilPallets = .90;
SetSlotOption( Pallets:MaxUtilPallets, AFTER_CHANGE, Pallets_Util_Diag );
MakeSlot( Pallets:Pallets_Util_Diag );
SetSlotOption( Pallets:Pallets_Util_Diag, ALLOWABLE_VALUES, Baixa, Normal, Alta );
Pallets:Pallets_Util_Diag = Alta;
MakeSlot( Pallets:UtilMéd );
Pallets:UtilMéd = 7.738;
SetSlotOption( Pallets:UtilMéd, AFTER_CHANGE, Pallets_Util_Diag );
MakeSlot( Pallets:UtilTotal );
Pallets:UtilTotal = 38.69;
MakeSlot( Pallets:NMinPallets );
Pallets:NMinPallets = 8;
MakeSlot( Pallets:NMaxPallets );
Pallets:NMaxPallets = 16;
MakeSlot( Pallets:Step );
Pallets:Step = 4;
SetSlotOption( Pallets:Step, WHEN_ACCESS, DefStep );
MakeSlot( Pallets:NLimPallets );
Pallets:NLimPallets = 16;

/*****
**** INSTANCE: VAGs
*****/
MakeInstance( VAGs, Transportadores );

/***** METHOD: AGV_Util_Diag *****/
MakeMethod( VAGs, AGV_Util_Diag, [] ).
If ( ( Self:UtilMéd / Self:NumVAGs ) < Self:Min_Allow_Util )
Then ( Self:Util_Diag = Baixa )
Else If ( Self:UtilMéd / Self:NumVAGs <= Self:MaxUtilVAGs )
Then ( Self:Util_Diag = Normal )
Else Self:Util_Diag = Alta );

/***** METHOD: AGV_Queue_Diag *****/
MakeMethod( VAGs, AGV_Queue_Diag, [] ).
If ( Self:TamMédFila < 1.0 )

```

```

Then ( Self:Diag_Fila = Curta )
Else If ( Self:TamMédFila <= Self:TamBuffer )
    Then ( Self:Diag_Fila = Normal )
    Else Self:Diag_Fila = Longa );

/***** METHOD: DefNumVAGs *****/
MakeMethod( VAGs, DefNumVAGs, []).
{
    If ( VAGs:NovoNumVAGs > VAGs:NumMaxVAGs )
        Then PostMessage( "Numero de VAGs deve ser menor ou igual a ".
            VAGs:NumMaxVAGs )
    Else {
        VAGs:Dif = VAGs:NumVAGs - VAGs:NovoNumVAGs;
        If ( VAGs:Dif > 0.0 )
            Then {
                For x From ( VAGs:NovoNumVAGs + 1 ) To ( VAGs:NovoNumVAGs + VAGs:Dif )

                    Do SetNthElem( VAGs:ListaEstadoVAGs, x, 2.0 );
                    VAGs:NumVAGs = VAGs:NovoNumVAGs;
                }
            Else {
                If ( VAGs:Dif < 0.0 )
                    Then {
                        For x From ( VAGs:NumVAGs + 1 ) To ( VAGs:NumVAGs + Abs( VAGs:Dif ) )

                            Do SetNthElem( VAGs:ListaEstadoVAGs,
                                x, 0.0 );
                        VAGs:NumVAGs = VAGs:NovoNumVAGs;
                    };
            };
    };
};

MakeSlot( VAGs:TamMédFila );
VAGs:TamMédFila = 0.074;
MakeSlot( VAGs:DesvioCompFila );
VAGs:DesvioCompFila = 1.17;
MakeSlot( VAGs:Min_Tam_Fila_na_Simul );
VAGs:Min_Tam_Fila_na_Simul = .00;
MakeSlot( VAGs:Máx_Tam_Fila_na_Simul );
VAGs:Máx_Tam_Fila_na_Simul = 4.00;
MakeSlot( VAGs:Util );
VAGs:Util = 3.25;
MakeSlot( VAGs:DesvioUtil );
VAGs:DesvioUtil = .32;
MakeSlot( VAGs:MinNumUtil );
VAGs:MinNumUtil = .00;
MakeSlot( VAGs:MáxNumUtil );
VAGs:MáxNumUtil = 4.00;
MakeSlot( VAGs:NumVAGs );
SetSlotOption( VAGs:NumVAGs, ALLOWABLE_VALUES, 1, 2, 3, 4 );
VAGs:NumVAGs = 4;
MakeSlot( VAGs:Min_Allow_Util );
SetSlotOption( VAGs:Min_Allow_Util, MINIMUM_VALUE, 0.0 );

```

```

SetSlotOption( VAGs:Min_Allow_Util, MAXIMUM_VALUE, 1.0 );
VAGs:Min_Allow_Util = .65;
SetSlotOption( VAGs:Min_Allow_Util, AFTER_CHANGE, AGV_Util_Diag );
MakeSlot( VAGs:MaxUtilVAGs );
SetSlotOption( VAGs:MaxUtilVAGs, MINIMUM_VALUE, 0.0 );
SetSlotOption( VAGs:MaxUtilVAGs, MAXIMUM_VALUE, 1.0 );
VAGs:MaxUtilVAGs = .90;
MakeSlot( VAGs:Util_Diag );
SetSlotOption( VAGs:Util_Diag, ALLOWABLE_VALUES, Baixa, Normal, Alta );
VAGs:Util_Diag = Normal;
MakeSlot( VAGs:Diag_Fila );
SetSlotOption( VAGs:Diag_Fila, ALLOWABLE_VALUES, Curta, Normal, Longa );
VAGs:Diag_Fila = Curta;
MakeSlot( VAGs:TamBuffer );
VAGs:TamBuffer = 5.0;
SetSlotOption( VAGs:TamBuffer, AFTER_CHANGE, AGV_Queue_Diag );
MakeSlot( VAGs:ListaEstadoVAGs );
SetValue( VAGs:ListaEstadoVAGs, 0.0, 0.0, 0.0, 0.0 );
MakeSlot( VAGs:NovoNumVAGs );
SetSlotOption( VAGs:NovoNumVAGs, ALLOWABLE_VALUES, 1, 2, 3, 4 );
VAGs:NovoNumVAGs = 4;
SetSlotOption( VAGs:NovoNumVAGs, AFTER_CHANGE, DefNumVAGs );
MakeSlot( VAGs:Dif );
VAGs:Dif = -1;
MakeSlot( VAGs:Pol_Despacho );
SetSlotOption( VAGs:Pol_Despacho, ALLOWABLE_VALUES, FIFO, MTPF, MNVR, MEBS );
VAGs:Pol_Despacho = FIFO;
MakeSlot( VAGs:MaxViagSCarga );
SetSlotOption( VAGs:MaxViagSCarga, MINIMUM_VALUE, 0.0 );
SetSlotOption( VAGs:MaxViagSCarga, MAXIMUM_VALUE, 1.0 );
VAGs:MaxViagSCarga = 0.8;
MakeSlot( VAGs:TamFila );
VAGs:TamFila = .08;
MakeSlot( VAGs:TamFilaTotal );
VAGs:TamFilaTotal = 0.37;
MakeSlot( VAGs:UtilMéd );
VAGs:UtilMéd = 3.246;
SetSlotOption( VAGs:UtilMéd, AFTER_CHANGE, AGV_Util_Diag );
MakeSlot( VAGs:UtilTotal );
VAGs:UtilTotal = 16.23;
MakeSlot( VAGs:UtilMédReal );
VAGs:UtilMédReal = 0.8115;
MakeSlot( VAGs:NumMaxVAGs );
VAGs:NumMaxVAGs = 4;

/*****
**** INSTANCE: Operação1
*****/
MakeInstance( Operação1, Operações );

/***** METHOD: Time_Operated_on_All_Machines *****/
MakeMethod( Operação1, Time_Operated_on_All_Machines, []
Self:Time_Operated_on_All_Machines = Self:OperX_on_Mach1 +

```

```

Self:OperX_on_Mach2 + Self:OperX_on_Mach3 + Self:OperX_on_Mach4 );
Operação1:Best_Time = 5;
SetValue( Operação1:List_of_Possibles_Machines, 1, 4, 2, 0, 0 );
MakeSlot( Operação1:Time_Operated_on_All_Machines );
Operação1:Time_Operated_on_All_Machines = 1970.01;

/*****
**** INSTANCE: Operação2
*****/
MakeInstance( Operação2, Operações );

/***** METHOD: Time_Operated_on_All_Machines *****/
MakeMethod( Operação2, Time_Operated_on_All_Machines, [],
Self:Time_Operated_on_All_Machines = Self:OperX_on_Mach1 +
Self:OperX_on_Mach2 + Self:OperX_on_Mach3 + Self:OperX_on_Mach4 );
Operação2:Best_Time = 5;
SetValue( Operação2:List_of_Possibles_Machines, 2, 4, 3, 0, 0 );
MakeSlot( Operação2:Time_Operated_on_All_Machines );
Operação2:Time_Operated_on_All_Machines = 2303.36;

/*****
**** INSTANCE: Operação3
*****/
MakeInstance( Operação3, Operações );

/***** METHOD: Time_Operated_on_All_Machines *****/
MakeMethod( Operação3, Time_Operated_on_All_Machines, [],
Self:Time_Operated_on_All_Machines = Self:OperX_on_Mach1 +
Self:OperX_on_Mach2 + Self:OperX_on_Mach3 + Self:OperX_on_Mach4 );
Operação3:Best_Time = 5;
SetValue( Operação3:List_of_Possibles_Machines, 3, 1, 2, 0, 0 );
MakeSlot( Operação3:Time_Operated_on_All_Machines );
Operação3:Time_Operated_on_All_Machines = 1885.5;

/*****
**** INSTANCE: Operação4
*****/
MakeInstance( Operação4, Operações );

/***** METHOD: Time_Operated_on_All_Machines *****/
MakeMethod( Operação4, Time_Operated_on_All_Machines, [],
Self:Time_Operated_on_All_Machines = Self:OperX_on_Mach1 +
Self:OperX_on_Mach2 + Self:OperX_on_Mach3 + Self:OperX_on_Mach4 );
Operação4:Best_Time = 5;
SetValue( Operação4:List_of_Possibles_Machines, 3, 2, 4, 0, 0 );
MakeSlot( Operação4:Time_Operated_on_All_Machines );
Operação4:Time_Operated_on_All_Machines = 1849.76;

/*****
**** INSTANCE: Operação5
*****/
MakeInstance( Operação5, Operações );

```

```

/***** METHOD: Time_Operated_on_All_Machines *****/
MakeMethod( Operação5, Time_Operated_on_All_Machines, []).
Self:Time_Operated_on_All_Machines = Self:OperX_on_Mach1 +
Self:OperX_on_Mach2 + Self:OperX_on_Mach3 + Self:OperX_on_Mach4 );
Operação5:Best_Time = 5;
SetValue( Operação5:List_of_Possibles_Machines, 4, 2, 1, 0, 0 );
MakeSlot( Operação5:Time_Operated_on_All_Machines );
Operação5:Time_Operated_on_All_Machines = 665;

/*****
**** INSTANCE: Operação6
*****/
MakeInstance( Operação6, Operações );

/*****
**** INSTANCE: Peça2
*****/
MakeInstance( Peça2, Peças );
SetValue( Peça2:ListaOper, 5, 4, 9, 2, 4, 0, 0, 0, 0, 0 );
SetValue( Peça2:Tempo_Processamento, 6, 14, 11, 7, 11, 0, 0, 0, 0, 0 );
Peça2:Tempo_Esperado_Passagem = 147;
Peça2:TFxSim = 135.52;
Peça2:Diag_Tempo_Passagem = Alta;
Peça2:ProdSim = 24.00;
Peça2:Meta_Prod = 24;
Peça2:Diag_de_Objetivos_Operac = Alcançado;
Peça2:TempoMédioTransp = 4.29;
Peça2:Tempo_Acumulado_Processamento = 49;
Peça2:TFxSimTotal = 672.21;
Peça2:ProdSimTotal = 120;
Peça2:TMédFxSim = 134.442;
Peça2:ProdMédSim = 24;

/*****
**** INSTANCE: Peça3
*****/
MakeInstance( Peça3, Peças );
SetValue( Peça3:ListaOper, 3, 4, 5, 6, 7, 0, 0, 0, 0, 0 );
SetValue( Peça3:Tempo_Processamento, 3, 4, 5, 6, 7, 0, 0, 0, 0, 0 );

/*****
**** INSTANCE: Peça4
*****/
MakeInstance( Peça4, Peças );
SetValue( Peça4:ListaOper, 4, 5, 6, 7, 10, 0, 0, 0, 0, 0 );
SetValue( Peça4:Tempo_Processamento, 6, 11, 15, 8, 4, 0, 0, 0, 0, 0 );
Peça4:Tempo_Esperado_Passagem = 132;
Peça4:TFxSim = 128.71;
Peça4:Diag_Tempo_Passagem = Alta;
Peça4:ProdSim = 18.00;
Peça4:Meta_Prod = 18;
Peça4:Diag_de_Objetivos_Operac = Alcançado;
Peça4:TempoMédioTransp = 3.90;

```

```

Peça4:Tempo_Acumulado_Processamento = 44;
Peça4:TFxSimTotal = 646.32;
Peça4:ProdSimTotal = 90;
Peça4:TMédFxSim = 129.264;
Peça4:ProdMédSim = 18;

```

```

/*****
**** INSTANCE: Peça5
*****/

```

```

MakeInstance( Peça5, Peças );
SetValue( Peça5:ListaOper, 5, 6, 7, 8, 9, 0, 0, 0, 0 );
SetValue( Peça5:Tempo_Processamento, .5, .6, .7, .8, .9, 0, 0, 0, 0 );

```

```

/*****
**** INSTANCE: Peça6
*****/

```

```

MakeInstance( Peça6, Peças );
SetValue( Peça6:ListaOper, 6, 7, 8, 9, 10, 0, 0, 0, 0 );
SetValue( Peça6:Tempo_Processamento, .6, .7, .8, .9, 1, 0, 0, 0, 0 );

```

```

/*****
**** INSTANCE: Peça7
*****/

```

```

MakeInstance( Peça7, Peças );
SetValue( Peça7:ListaOper, 7, 8, 9, 10, 0, 0, 0, 0, 0 );

```

```

/*****
**** INSTANCE: Peça8
*****/

```

```

MakeInstance( Peça8, Peças );
SetValue( Peça8:ListaOper, 2, 1, 3, 5, 3, 2, 0, 0, 0 );

```

```

/*****
**** INSTANCE: Peça9
*****/

```

```

MakeInstance( Peça9, Peças );
SetValue( Peça9:ListaOper, 2, 1, 3, 5, 3, 2, 0, 0, 0 );

```

```

/*****
**** INSTANCE: Peça10
*****/

```

```

MakeInstance( Peça10, Peças );
SetValue( Peça10:ListaOper, 2, 1, 3, 5, 3, 2, 0, 0, 0 );

```

```

/*****
**** INSTANCE: Peça11
*****/

```

```

MakeInstance( Peça11, Peças );
SetValue( Peça11:ListaOper, 2, 1, 3, 5, 3, 2, 0, 0, 0 );

```

```

/*****
**** INSTANCE: Peça12
*****/

```

```

MakeInstance( Peça12, Peças );
SetValue( Peça12:ListaOper, 2, 1, 3, 5, 3, 2, 0, 0, 0, 0 );
Peça12:ProdSim = 5;

```

```

/*****
**** INSTANCE: Peça13
*****/

```

```

MakeInstance( Peça13, Peças );
SetValue( Peça13:ListaOper, 9, 6, 8, 6, 8, 0, 0, 0, 0, 0 );
SetValue( Peça13:Tempo_Processamento, 3, 3, 15, 5, 15, 0, 0, 0, 0, 0 );
Peça13:Tempo_Esperado_Passagem = 123;
Peça13:Meta_Prod = 12;
Peça13:ProdSim = 12.00;
Peça13:TFxSim = 120.61;
Peça13:Diag_de_Objeticivos_Operac = Alcançado;
Peça13:Diag_Tempo_Passagem = Alta;
Peça13:TempoMédioTransp = 3.72;
Peça13:Tempo_Acumulado_Processamento = 41;
Peça13:TFxSimTotal = 624.72;
Peça13:ProdSimTotal = 60;
Peça13:TMédFxSim = 124.944;
Peça13:ProdMédSim = 12;

```

```

/*****
**** INSTANCE: Peça14
*****/

```

```

MakeInstance( Peça14, Peças );
SetValue( Peça14:ListaOper, 4, 2, 1, 3, 5, 3, 2, 0, 0, 0 );

```

```

/*****
**** INSTANCE: Peça15
*****/

```

```

MakeInstance( Peça15, Peças );
SetValue( Peça15:ListaOper, 2, 1, 3, 5, 3, 2, 0, 0, 0, 0 );

```

```

/*****
**** INSTANCE: Peça16
*****/

```

```

MakeInstance( Peça16, Peças );
SetValue( Peça16:ListaOper, 2, 1, 3, 5, 3, 2, 0, 0, 0, 0 );

```

```

/*****
**** INSTANCE: Peça17
*****/

```

```

MakeInstance( Peça17, Peças );
SetValue( Peça17:ListaOper, 2, 3, 5, 9, 5, 1, 0, 0, 0, 0 );
SetValue( Peça17:Tempo_Processamento, 18, 4, 4, 6, 8, 12, 0, 0, 0, 0 );
Peça17:Tempo_Esperado_Passagem = 156;
Peça17:Meta_Prod = 24;
Peça17:ProdSim = 24.00;
Peça17:TFxSim = 145.37;
Peça17:Diag_de_Objeticivos_Operac = Alcançado;
Peça17:Diag_Tempo_Passagem = Alta;

```



```

Peça17:TempoMédioTransp = 4.09;
Peça17:Tempo_Acumulado_Processamento = 52;
Peça17:TFxSimTotal = 692.91;
Peça17:ProdSimTotal = 120;
Peça17:TMédFxSim = 138.582;
Peça17:ProdMédSim = 24;

```

```

/*****
**** INSTANCE: Peça18
*****/

```

```

MakeInstance( Peça18, Peças );
SetValue( Peça18:ListaOper, 2, 6, 8, 9, 5, 6, 0, 0, 0, 0 );
SetValue( Peça18:Tempo_Processamento, 6, 13, 8, 7, 12, 16, 0, 0, 0, 0 );
Peça18:Tempo_Esperado_Passagem = 186;
Peça18:Meta_Prod = 30;
Peça18:ProdSim = 30.00;
Peça18:TFxSim = 155.20;
Peça18:Diag_de_Objetivos_Operac = Alcançado;
Peça18:Diag_Tempo_Passagem = Aceitável;
Peça18:TempoMédioTransp = 3.97;
Peça18:Tempo_Acumulado_Processamento = 62;
Peça18:TFxSimTotal = 791.61;
Peça18:ProdSimTotal = 150;
Peça18:TMédFxSim = 158.322;
Peça18:ProdMédSim = 30;

```

```

/*****
**** INSTANCE: Peça19
*****/

```

```

MakeInstance( Peça19, Peças );
SetValue( Peça19:ListaOper, 2, 3, 0, 0, 0, 0, 0, 0, 0, 0 );

```

```

/*****
**** INSTANCE: Peça20
*****/

```

```

MakeInstance( Peça20, Peças );
SetValue( Peça20:ListaOper, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 );

```

```

/*****
**** INSTANCE: Operação7
*****/

```

```

MakeInstance( Operação7, Operações );

```

```

/*****
**** INSTANCE: Operação8
*****/

```

```

MakeInstance( Operação8, Operações );

```

```

/*****
**** INSTANCE: Operação9
*****/

```

```

MakeInstance( Operação9, Operações );

```

```

/*****
**** INSTANCE: Operação10
*****/
MakeInstance( Operação10, Operações );

/*****
**** INSTANCE: SimÑSim1
*****/
MakeInstance( SimÑSim1, Menu );

/***** METHOD: Simular_Novamente *****/
MakeMethod( SimÑSim1, Simular_Novamente, []
{
Pallets:NPallets = Pallets:NPallets + Pallets:Step;
SalvaParSimAnálise( );
PostMessage( "Voltando a Simular com um incremento no N° de Pallets." );
Execute( Global:Exec_Simul_File, Global:Exec_Simul_Model );
HideWindow( DefParamAnálise );
ShowWindow( Análise );
PostMessage( "Simulação Terminada" );
} );

/***** METHOD: Não_Simular *****/
MakeMethod( SimÑSim1, Não_Simular, []
{
PostMessage( "Usuário Encerrou Processo Interativo Simulação/Análise." );
HideWindow( DefParamAnálise );
ShowWindow( Análise );
} );
SimÑSim1.X = 318;
SimÑSim1.Y = 239;
SetValue( SimÑSim1.Choices, Simular_Novamente, Não_Simular );

/*****
**** INSTANCE: SimÑSim2
*****/
MakeInstance( SimÑSim2, Menu );

/***** METHOD: Não_Simular *****/
MakeMethod( SimÑSim2, Não_Simular, []
{
PostMessage( "Usuário Encerrou Processo Interativo Simulação/Análise." );
HideWindow( DefParamAnálise );
ShowWindow( Análise );
} );

/***** METHOD: Simular_Novamente *****/
MakeMethod( SimÑSim2, Simular_Novamente, []
{
Pallets:NLimPallets = Pallets:NPallets - Pallets:Step;
Pallets:NPallets = Pallets:NLimPallets;
Sistema:TMédCPSimAnt = Sistema:TMédCPSim;
Máquinas:TamBufferEnt += 1;

```

```

SalvaParSimAnálise( );
PostMessage( " Voltando a Simular com um aumento nos Buffers e uma redução nos Pallets." );
Execute( Global:Exec_Simul_File, Global:Exec_Simul_Model );
HideWindow( DefParamAnálise );
ShowWindow( Análise );
PostMessage( "Simulação Terminada" );
} );
SimÑSim2:X = 326;
SimÑSim2:Y = 34;
SetValue( SimÑSim2:Choices, Simular_Novamente, Não_Simular );

/*****
**** INSTANCE: SimÑSim3
*****/
MakeInstance( SimÑSim3, Menu );

/***** METHOD: Simular_Novamente *****/
MakeMethod( SimÑSim3, Simular_Novamente, []).
{
  Máquinas:TamBufferEnt = Máquinas:TamBufferEnt + 1;
  Sistema:TMédCPSimAnt = Sistema:TMédCPSim;
  SalvaParSimAnálise( );
  PostMessage( "Voltando a Simular com tamanho do Buffer de entrada incrementado" );
  Execute( Global:Exec_Simul_File, Global:Exec_Simul_Model );
  HideWindow( DefParamAnálise );
  ShowWindow( Análise );
  PostMessage( "Simulação Terminada" );
} );

/***** METHOD: Não_Simular *****/
MakeMethod( SimÑSim3, Não_Simular, []).
{
  PostMessage( "Usuário Encerrou Processo Interativo Simulação/Análise" );
  HideWindow( DefParamAnálise );
  ShowWindow( Análise );
} );
SimÑSim3:X = 320;
SimÑSim3:Y = 227;
SetValue( SimÑSim3:Choices, Simular_Novamente, Não_Simular );

/*****
**** INSTANCE: SimÑSim4
*****/
MakeInstance( SimÑSim4, Menu );

/***** METHOD: Não_Simular *****/
MakeMethod( SimÑSim4, Não_Simular, []).
{
  PostMessage( "Usuário Encerrou Processo Interativo Simulação/Análise" );
  HideWindow( DefParamAnálise );
  ShowWindow( Análise );
} );

```

```

/***** METHOD: Simular_Novamente *****/
MakeMethod( SimNSim4, Simular_Novamente, []).

```

```

{
  VAGs:NovoNumVAGs += 1;
  Pallets:NLimPallets = Pallets:NMaxPallets;
  Sistema:TMédCPSimAnt = Sistema:TMédCPSim;
  SalvaParSimAnálise( );
  PostMessage( " Voltando a Simular com nº de VAGs incrementado." );
  Execute( Global:Exec_Simul_File, Global:Exec_Simul_Model );
  HideWindow( DefParamAnálise );
  ShowWindow( Análise );
  PostMessage( "Simulação Terminada" );
} );
SimNSim4:X = 371;
SimNSim4:Y = 295;
SetValue( SimNSim4:Choices, Simular_Novamente, Não_Simular );

```

```

/*****
*** INSTANCE: SimNSim5
*****/

```

```

MakeInstance( SimNSim5, Menu );

```

```

/***** METHOD: Não_Simular *****/
MakeMethod( SimNSim5, Não_Simular, []).

```

```

{
  PostMessage( "Usuário Encerrou Processo Interativo Simulação/Análise" );
  HideWindow( DefParamAnálise );
  ShowWindow( Análise );
} );

```

```

/***** METHOD: Simular_Novamente *****/
MakeMethod( SimNSim5, Simular_Novamente, []).

```

```

{
  Pallets:NLimPallets = Pallets:NPallets - Pallets:Step;
  Pallets:NPallets = Pallets:NLimPallets;
  Máquinas:TamBufferEnt += 1;
  Sistema:TMédCPSimAnt = Sistema:TMédCPSim;
  SalvaParSimAnálise( );
  PostMessage( " Voltando a Simular com um aumento nos Buffers e uma redução nos Pallets." );
  Execute( Global:Exec_Simul_File, Global:Exec_Simul_Model );
  HideWindow( DefParamAnálise );
  ShowWindow( Análise );
  PostMessage( "Simulação Terminada" );
} );
SimNSim5:X = 345;
SimNSim5:Y = 292;
SetValue( SimNSim5:Choices, Simular_Novamente, Não_Simular );

```

```

/*****
*** INSTANCE: SimNSim7
*****/

```

```

MakeInstance( SimNSim7, Menu );

```

```

/***** METHOD: Não_Simular *****/
MakeMethod( SimNSim7, Não_Simular, []).
{
  PostMessage( "Usuário Encerrou Processo Interativo Simulação/Análise" );
  HideWindow( DefParamAnálise );
  ShowWindow( Análise );
} );

/***** METHOD: Simular_Novamente *****/
MakeMethod( SimNSim7, Simular_Novamente, []).
{
  VAGs:NovoNumVAGs += 1;
  Sistema:TMédCPSimAnt = Sistema:TMédCPSim;
  SalvaParSimAnálise( );
  PostMessage( "Voltando a Simular com nº de VAGs incrementado." );
  Execute( Global:Exec_Simul_File, Global:Exec_Simul_Model );
  HideWindow( DefParamAnálise );
  ShowWindow( Análise );
  PostMessage( "Simulação Terminada" );
} );
SimNSim7:X = 337;
SimNSim7:Y = 293;
SetValue( SimNSim7:Choices, Simular_Novamente, Não_Simular );

```

REFERÊNCIAS BIBLIOGRÁFICAS

- ADELSBERGER, H.H. et alii., "Rule based object-oriented simulation systems". Intelligent Simulation Systems, SCS, Simulation Series, 107-112, Jan, 1986.
- ADELSBERGER, H.H. e NEUMANN, G., "Goal oriented simulation modeling using PROLOG", Proc. of the 1985 SCS Conf. on Modeling. and Simulation On Micro-Computers, San Diego, Cal, 42-47, 1985
- AMADOR, R.M.S.T. e PIMENTÃO, J.P.B., "Representação de conhecimento através de "frames", Univ. Nova Lisboa, Dep. Informática, Portugal, 1988
- ARBEL, A. e SEIDMANN, A. " Performance Evaluation of FMS ". IEEE Transactions on Systems, Man and Cybernetics, July-August, 1984, pp. 606-617.
- AZZONE, G. e BERTELÈ, U., "Measuring the economic effectiveness of flexible automation: a new approach", International Journal of Production Research., v. 27, 5, 735-746, 1989.
- BANKS. J. e ARSON, J.S. Discrete-Event System Simulation. Prendice-Hall, Englewood Cliffs, NJ, 1984.
- BARAD, M. e SIPPER, D., "Flexibility in manufacturing systems definitions and Petri net modeling", International Journal of Production Research, 26, 237-248, 1988.
- BENJAAFAR. S., "Intelligent Simulation for Flexible Manufacturing Systems: an Integrated Approach., Computers and Industrial Engineering, V.22, N° 3, 297-311, 1992.
- BEVANS, J.P., "First chose an FMS simulator", Americam Machinist, 126, May, 143-145, 1982.

- BOZER, Y.A., "Guided Vehicle Systems: Information/Control system implications of alternative design and operation strategies", In: *Advanced Information Technologies for Industrial Material Flow Systems*, Ed. S.Y. Nof and C.L. Moodie, 1989.
- BRILL, P.H. e MANDELBAUM, M., "On measures of flexibility in manufacturing systems", *International Journal of Production Research*, 27, 747-756, 1989
- BROWNE, J. et alii., "The development of a FMS design procedure", in *Proceedings of the First ORSA/TIMS Special Interest Conference on FMS*, K. Stecke & R. Suri (ed.), Ann Harbor, Michigan, 1984.
- BUZACOTT, J.A., "The fundamental principles of flexibility in manufacturing systems", *Proc. of the First Int. Conf.*, Brighton UK, Elsevier, Amsterdam, 1982
- BUZACOTT, J.A. e MANDELBAUM, M., "Flexibility and Productivity in manufacturing systems", in *Proc. of the IIE Conf.*, Chiago, IL., Ind. Eng. Man. Press, 404-413, 1985
- BUZACOTT, J.A. e SHANTICUMAR, J.A., "Models for understanding FMS, *AIIE Transactions*, 12, 339-349, 1986.
- BUZACOTT, J.A. e YAO, D.D., "FMS a review of analitical Models", *Management Science*, 32, 890-904, 1986.
- CARLIER, J. , "Simulation as planning tool for an FMS", in *Proc. of 4th International Conference of Simulation in Manufacturing*, 231-238, 1988.
- CARRIE, A., "The role of simulation in FMS", in: A. Kusiak (ed.), *Flexible Manufacturing Systems: Methods and Studies*. Norht-Holland, Amsterdam, 191-208, 1986.
- CARRIE, A., *Simulation of Manufacturing Systems*. John Wiley & Sons Ltd., 1988, London.
- CARRIE, A.S. e PETSOPOULOS, A.C., "Operation Sequencing in a FMS", *Robotica*, 3, 259-264, 1985.
- CHATTERJEE, A. et alii., "Manufacturing Flexibility: Models and measurements", in *Proc. of the First ORSA/TIMS Special Interest Conf. in FMS*, K.E. STECKE e R. SURI (ed.), Ann Arbor, Mich., Elsevier, Ams., 1984.

- CHANDRA, J. e SCHALL, S.O. "Economic justification of FMS's using the Leontief input-output model". *The Engineering Economist*. Vol.34, N: 01, 1988.
- CHEN, C.S. e SARIN, S.C., "A mathematical model for manufacturing system selection, in: A.Kusiak (ed.), *Flexible Manufacturing Systems: Methods and Studies*. North-Holland, Amsterdam, 03-25, 1986.
- CHENG, T.C.E., "Simulation of Flexible Manufacturing Systems", *Simulation*, 45, Dec., 299-302, 1985.
- CHORAFAS, D. N., *Knowledge Engineering*, Van Nostrand Reinhold Edition, New York, 1990.
- CHRISTY, D.P. e WATSON, H.J., "The application of simulation: A survey of a industry practice", *Interfaces*, 13, Oct, 47-55, 1983.
- COSTA, R.S. e JARDIM, E.G.M. " Simulação no Gerenciamento da Produção". *Informativo do INT*, pg. 13-24, v. 17, 34, jul/dez. 1985.
- CO, H.C., JAW, T.J., Sequencing in manufacturing systems and other short queue length systems, *Journal of Manufacturing Systems*, 7, 1-9, 1988
- CUNHA, C.J.C.A. " Produção e estratégia ". *Anais do IX ENEGEP*. Porto Alegre, Set. 1989.
- CUNHA, C.J.C.A. "Aspectos estratégicos do CIM". *Anais o IX ENEGEP*. Porto Alegre, Set. 1989
- CZIULIK, C. e ROSA, E. "O CIM como elemento do planejamento estratégico das organizações". *Anais do VIII ENEGEP*. São Paulo, 1988.
- DUBOIS, D. e STECKE, K. E., "Using Petri nets to represent production processes, *Proc. 22nd IEEE Conference in control*, 1983.
- EGBELU, P.J., e TANCHOCO, J.M.A., "Characterization of the automatic guided vehicle dispatching rules, *International Journal of Production Research*, 22, 359-374, 1984.
- EGBELU, P. J., "Establishment of economic production rate, production batch size and production sequence in manufacturing systems with flexible routing". *European Journal of Operational Research*, 67, 358-372, 1993.
- ELMARAGHY, H.A. e HO, N.C., "A flexible system for computer control of manufacturing", *Computer in Mechanical Engineering*, 1, 53-63, 1982

- ERKES, K. e SCHMIDT, H. "Flexible Fertigung". VDI- Zeitschrift, pg. 577-591, 1987.
- FIKES, R. e KEHLER, T., "The role of frame-based representation in reasoning", Commun. of the ACM, v 28, 9, 904-920, 1985.
- FISCHLER, M.A. e FIRSCHEIN, O., Intelligence - The Eye, the Brain, and the Computer, Addison-Wesley, 1987.
- FREITAS, P.J., "Uso de sistemas inteligentes de simulação no gerenciamento de sistemas flexíveis de manufatura", Anais do X ENEGEP, Belo Horizonte, 1990.
- FUTO, M.S. e SZEREDLI, J., "A discrete simulation system based on artificial intelligence methods", In Discrete Simulation and Related Fields, A.JAVOR (ed), North-Holland, 135-150 1982.
- GAINES, B.R. e SHAW, M.L.G., "Expert systems and Simulation", in Proc. of the Conf. on Artif. Intell., Graphics and Simulation, G. BIRTWISTLE (ed.), San Diego, Cal., 95-101, 1985.
- GARRET, S.E., "Strategy first: a case in FMS justification", in Proc. of the second ORSA/TIMS Conf. on FMS, Ann Arbor, Mich., K.E. STECKE e R.SURI (ed.), Elsevier, Amsterdam, 17-30, 1986.
- GERWIN, D. "Do's and don'ts of computerized manufacturing". Harvard Business Review, pg. 107-116, 1982.
- GERWIN, D. e TARONDEAU, J.C., "Internacional comparisons of manufacturing flexibility", In Managing International Manufacturing, K.FERDOWS (ed), Elsevier, Amsterdam, 169-185, 1989.
- GOLDHAR, J. e JELINEK, M. "Plan for Economies of Scope". Harvard Business Review, 61(6), 1983, pp. 141-148.
- GREENWOOD, N. R. Implementing Flexible Manufacturing Systems. 1988. John Wiley & Sons, Inc., New York.
- GROOVER, M.P. Automation, Production Systems and Computer Aided Manufacturing. Prentice-Hall, Inc. Englewood Cliffs, New Jersey, 1980.
- GUPTA, Y.P. e GOYAL, S. "Flexibility of Manufacturing Systems: Concepts and Measurements". European Journal of Operational Research, 43, pp. 119-135, 1989.

- GUPTA, Y.P. e GOYAL, S. " Flexibility trade-offs in a random FMS: A simulation study. *International Journal of Production Research*. V-30, Nº 3, March, 527-557, 1992.
- HARMON, P. e KING, D., *Sistemas Especialistas*, Ed. Campus, 1988
- HAYES, R.H. e JAYCUMAR, R. " Manufacturing's Crisis: New technologies, Obsolete Organizations", *Harvard Business Review*, 66(5), 77-85, 1988.
- HAYES-ROTH, F. et al. *Building Expert Systems*. Addison-Wesley Publishing Co., 1983.
- HAYES, R. H. e WHEELWRIGHT, S. C. " Dinâmica dos Ciclos de Vida Processo-Produto ". *Biblioteca Harvard de Administração de Empresas*. n. 4, vol. 8. Ed. Abril.
- HAYES, R.H. e WHEELWRIGHT, S.C. *Restoring our Competitive Edge*. New York, J. Wiley, 1984.
- HENRIKSEN, J.O., "The integrated simulation environment (Simulation software of the 1990s)", *Operations Research*, 31, Nov-Dec, 1053-1073, 1983.
- HILL, T., *Manufacturing Strategy - The Strategic Management of the Manufacturing Function*, Macmillan Education LTD, London, 1985.
- HO, Y. C., "A survey of the discrete event dynamic systems", *Annals of Operations Research*, 3, 1985.
- HURRION, R. D., "Intelligent Visual Interactive Modelling", *European Journal of Operational Research*, 54, 349-356, 1991.
- INTELLICORP, *KAPPA-PC 2.0 User's Manual*, Melo Park, Cal, 1992.
- INTELLICORP, *KEE , User's Manual*, Melo Park, Cal, 1985.
- JAIKUMAR, R., "Flexible Manufacturing Systems : A managerial perspective", Wp #1-784-078, Harvard Business School, Boston, Mass., Jan, 1984
- KAPLAN, R. S. " Must CIM be Justified by Faith Alone? ". *Harvard Business Review*, 64(2), 1986, pp. 87-95.
- KIRAN, A. S. et al., "An integrated simulation approach to design of flexible manufacturing systems", *Simulation*, Feb., 47-52, 1989.

- KLAHR, P., et alii., "Rule oriented Simulation", Proc. of the 1980 International Conference on Cybernetics and Society, IEEE, Cambridge, Mass., Oct., 350-354, 1980.
- KUSIAK, A., "Flexible Manufacturing Systems: A structural approach", International Journal of Production Research, 23, 1057-1073, 1985.
- KUSIAK, A. e CHEN, M., "Expert systems for planning and scheduling manufacturing systems", European Journal of Operational Research, 34, 113-130, 1988.
- LAW, A.M. e KELTON, W.D. Simulation Modeling and Analysis. McGraw-Hill, NY, 1991.
- LAW, A.M. " Introduction to Simulation: A Powerful Tool for Analyzing Complex Manufacturing Systems". Industrial Engineering, pg 46-62, Maio 1986.
- LEUNG, L. C., MAHESHWARI, S. K. e MILLER, W. A., "Concurrent part routing and tool allocation in FMS with material handling considerations", Working Paper, Department of Industrial and Management Systems Engineering, U.S.F., Tampa, USA, 1990.
- LEUNG, L. C. e TANCHOCO, J. M. A. , "Multiple machine replacement within an integrated framework, The Engineering Economist, 32, 89 - 114, 1987.
- MAHESHWARI, S. K. , "Planning and Operational control of FMS with material handling considerations", Ph.D. Dissertation, Department of Industrial and Management Systems Engineering, U.S.F., Tampa, USA, 1992.
- MARTIN, D.L. e MUSSELMAN, K.J., "Simulation in the life cycle of FMS", in Proceedings of the First ORSA/TIMS Special Conference on FMS, K.E. STECKE & R. SURI (ed.), Ann Arbor, Mich., 154-167, 1984.
- MEDEIROS, D.J. e SADOWSKI, R.P., "Simulation of robotic manufacturing cells: a modular approach", Simulation, 40, 03-12, 1983
- MISHRA, P. K. et alii, "Simulation studies of flexible manufacturing systems", in Proceedings of 4th International Conference of Simulation in Manufacturing, 119-128, 1988.
- MONTAZERI, M. e VAN WASSENHOVE, L.N., "Analysis of scheduling rules for FMS", Int. J. of Prod. Res., 28, 785-802, 1990.

- NOF, S. Y., BARASH, M., SOLBERG, J.J., "Operational control of item flow in versatile manufacturing systems", *International Journal of Production Research*, 17, 479-489, 1979.
- NELSON, C.A. "A scoring model for FMS project selection". *European Journal of Operational Research*, N: 24, 1986.
- O'KEEFE, R., "Simulation and Expert Systems - A taxonomy and some examples", *Simulation*, 46, Jan, 10-16, 1986
- PENGDEN, C.D. et alii., *Introduction to Simulation Using SIMAN*, McGraw-Hill, New Jersey, 1990
- PRIMROSE, P.L., e LEONARD, R., "Establish the viability of FMS", *The FMS Magazine*, 3, 114-116, 1985.
- RACHAMADUGU, R.V. e STECKE, K.E., "Classification and Review of FMS Scheduling Procedures", Working Paper, Graduate School of Business Administration, The University of Michigan, Ann Arbor, 1989.
- RADHARAMANAN, R. "Flexible Manufacturing Systems and Factories of the Future". *Seminário de Automação Industrial*, UFRGS, Set. 1989
- RAMAN, N. et alii., "Real time scheduling of automated manufacturing center", *European Journal of Operational Research*, 40, 222-224, 1989
- SABUNCUOGLU, I. e HOMMERTZHEIM, D.L. "Computer usage in manufacturing systems". *Computers and Industrial Engineering*. Vol. 15, N° 1-4, pg. 1-7, 1988.
- SABUNCUOGLU, I. e HOMMERTZHEIM, D.L. "Experimental investigation of FMS machine and AGV scheduling rules against the mean flow-time criterion". *European Journal of Operational Research*, V-30, n° 7, 1617-1635, 1992.
- SCHRIBER, T.J. e STECKE, K.E., "Machine utilizations achieved using balanced FMS production ratios in a simulation setting", *Annals of Productions Research*, 32, 579-595, 1988.
- SCHROER, B.J. e TSENG, F.T., "Modelling Complex manufacturing Systems Using Discret Event Simulation", *Computers and Industrial Engineering*, Vol 14, 4, pg 455-464, 1988.
- SCOTT, A. C. et alii., *A Practical Guide to Knowledge Acquisition*, Addison-Wesley Company, Inc. USA, 1991.

- SETHI, A.K. e SETHI, S.P., "Flexibility in Manufacturing: a survey", *The International Journal of FMS*, 2, 289-328, 1990.
- SHANNON, R.E., "Knowledge based simulation techniques for manufacturing", *International Journal of Production Research*, 3, 953-973, 1988.
- SHANNON, R.E. et alli " Expert system and simulation". *Simulation*, pg.275-284, jun. 1985.
- SKINNER, W. *Manufacturing: The Formidable Competitive Weapon*, New York, J. Wiley, 1985.
- SOLBERG, J.J., "Optimal design and control of computerized manufacturing systems", *Proc. of AIIE Systems Eng. Conference*, Boston, 137-147, 1976.
- SOLBERG, J.J., "CAN-Q User's guide", *Purdue University, School of IE*, 1980.
- SON, Y.K. e PARK, C.S. "An economic avaluation model for advanced manufacturing systems". *The Engineering Economist*. Vol. 34, 1, 1988.
- STECKE, K.E., "Design, planning, scheduling and control problems of FMS", *Annals of Operations. Research*, 3, 51-60, -1985.
- STECKE, K.E., "A hierarchical aproach to solving machining grouping and loading problems of FMS", *European Journal of Operational Research.*, 24, 369-378, 1986.
- STECKE, K.E. e SOLBERG, J.J., "Loading and control polices for FMS", *International Journal of Production Research*, 19, 481-490, 1981.
- SLACK, N., "The flexibility of Manufacturing systems", *International Journal of Operational and Production Management*, V.7, 4, 35-45, 1987
- SURI, R., "An overview of evaluative models for Flexible Manufacturing Systems", *Annals of Operations Research*, 3, 61-69, 1985.
- SURI, R. e DILLE, J. W., "A technique for on-line sensitivity analysis of flexible manufacturing systems", *Annals of Operations Research*, 3, 1985.
- SURI, R. e HILDEBRANT, R. R. , "Modeling Flexible Manufacturing Systems using mean value analysis", *Journal of Manufacturing Systems*, 3, 27, 1984
- SURI, R. e WHITNEY, C., " Decision support requirements in flexible manufacturing", *Journal of Manufacturing Systems*, 3, 185-193, 1984.

- SWAMIDASS, P.M. e NEWELL, W.T. "Manufacturing Strategy, Enviornmental Uncertainty and Performance: A Path Analytic Model ". Management Science 33(4), pp. 509-524, 1987.
- TAGUCHI, G., "System of Experimental Design", Vol 1-2, Kraus International Publications, New York, USA, 1987
- TALAVAGE, J. e HANNAM, R.G., Flexible Manufacturing Systems in Praticce, Applications, Design and Simulation., Marcel Dekker Inc. New York, 1988.
- TALAVAGE, J. e SHOLDHAN, R., "A combined AI-Simulation aproach for designing manufacturing systems", Proceedings. of 1987 SAE Int. Congress, Detroit, Feb, 1987.
- VIALI, L., Simulação Aplicada a Manufatura Flexível, Tese de Mestrado não publicada, Universidade Federal SC-CTC-EPS, 1991.
- VINAYAGAMOORTHY, R. e MITAL, A. "Economic feasibility of a robot installation". The Engineering Economist. Vol. 32, 3, 1987.
- WAHAB, F.A.F.A., An expert system for the design os FMS's using simulation analyses. Tese de doutorado. University of Alabama, 1986.
- WANG, H. , "An experimental analysis of the flexible manufacturing system (FMS), in: A.Kusiak (ed.), Flexible Manufacturing Systems: Methods and Studies. Norht-Holland, Amsterdam, 319-339, 1986.
- WALTERMAN, D.A., A Guide to Expert Systems. Addison-Wesley Publishing Co.. 1986
- WHITNEY, A. e SURI, R., "Algorithm for parts and machines selection in FMS", Annals of Operations. Research, 3, 95-117, 1985.
- WICK, C., "Advances in Machining Centers", Manufacturing Engineering, Oct.,24-32, 1987.
- WILHELM et alii., "Flexible Manufacturing Systems: A Review of Modeling Approches for Desing, Justification and Operation, in: A.Kusiak (ed.), Flexible Manufacturing Systems: Methods and Studies. Norht-Holland, Amsterdam, 03-25, 1986.

- WILHELM, W. E. e SHIN, H., "Effectiveness of alternate operations in a Flexible Manufacturing System, *International Journal of Production Research*, 23, 65-79, 1985.
- WINSTON, P.H., *Artificial Intelligence*, Reading, Mass.: Addison Wesley, 1984.
- WU, S.Y.D., WYSK, R.A., "An application of discrete event simulation to on-line control and scheduling in flexible manufacturing", *International Journal of Production Research*, 27, 1603-1623, 1989.
- YIM, D. S. e LINN, R. J., "Push and pull rules for dispatching AGV's in a FMS", *International Journal of Production Research*, 31, 43-57, 1993.
- ZEESTRATEM, M.J., "The look ahead dispatching procedure", *International Journal of Production Research*, 28, 369-384, 1990.
- ZELENOVIC, D.M., "Flexibility, a condition for effective production systems", *International Journal of Production Research*, V 20, 3, 319-337, 1982
- ZISK, B. I., " Flexibility is a Key to Automated Material Transport System for Manufacturing Cells", *Industrial Engineering*, Nov., 52-60, 1983.
- ZHUANG, L. e HINDI, K. S. "Approximate decomposition for closed queueing network models of FMS with block-and-wait and state-dependent routing mechanism", *European Journal of Operational Research*, 67, 3, 373-387, 1993.